

AN INVESTIGATION OF THE USE OF MICRO COMPUTERS
TO AID THE PILOT IN THE SOLUTION OF THE
BALLISTICS PROBLEM

James Phillip Pennell

LIBRARY
Postgraduate School
Monterey, California 93940

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

AN INVESTIGATION OF THE USE OF MICRO COMPUTERS
TO AID THE PILOT IN THE SOLUTION OF THE
BALLISTICS PROBLEM

by

James Phillip Pennell

June 1975

Thesis Advisor:

U. R. Kodres

Approved for public release; distribution unlimited.

T168323

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) An Investigation of the Use of Micro Computers to Aid the Pilot in the Solution of the Ballistics Problem		5. TYPE OF REPORT & PERIOD COVERED 1975 Master's Thesis; June
7. AUTHOR(s) James Phillip Pennell		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1975
		13. NUMBER OF PAGES 122
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Micro computer, ballistics problem, air to ground unguided weapons, air to air gunnery		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The advent of jet fighter/attack aircraft has resulted in the need for additional devices to aid the pilot in the accurate delivery of unguided weapons. The current generation of attack aircraft has met this need with the installation of minicomputers and additional sensor devices such as doppler radar and inertial navigation systems.		

This paper demonstrates a low cost method for improving the delivery accuracy of those aircraft which do not have such devices installed. A comparison is made between delivery accuracy with and without the proposed modification.

An Investigation of the Use of Micro Computers to Aid the
Pilot in the Solution of the Ballistics Problem

by

James Phillip Pennell
Major, United States Marine Corps
B.S., Oklahoma State University, 1972

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1975

ABSTRACT

The advent of jet fighter/attack aircraft has resulted in the need for additional devices to aid the pilot in the accurate delivery of unguided weapons. The current generation of attack aircraft has met this need with the installation of minicomputers and additional sensor devices such as doppler radar and inertial navigation systems.

This paper demonstrates a low cost method for improving the delivery accuracy of those aircraft which do not have such devices installed. A comparison is made between delivery accuracy with and without the proposed modification.

TABLE OF CONTENTS

I.	INTRODUCTION.....	7
II.	UNGUIDED WEAPONS DELIVERY USING FIXED SIGHT AIRCRAFT	9
	A. DEFINITIONS.....	9
	B. PROBLEM DEFINITION.....	11
	C. PILOT SOLUTION FOR AIR TO GROUND DELIVERY.....	14
	D. AIR TO AIR GUNNERY.....	19
III.	PROPOSED SOLUTION.....	20
	A. COMPONENTS.....	20
	1. Control Stick Sense Switch.....	22
	2. Portable Micro Computer with Auxiliary Memory	22
	3. Air Data Computer.....	22
	4. Loft Bombing Computer.....	22
	5. Cockpit Keyboard.....	23
	6. Micro Computer.....	23
	7. Gunsight.....	23
	8. Airborne Missile Control System.....	23
	9. Angle of Attack Sensor.....	24
	B. DESCRIPTION OF USE.....	24
	1. Air to Ground Mode.....	26
	2. Air to Air Mode.....	27
IV.	ANALYSIS AND TEST.....	28
	A. AIR TO GROUND	28
	1. General Considerations.....	28
	2. Tests of the Proposed Solution.....	30
	a. Static Accuracy.....	30
	b. Real Time Accuracy.....	34
	c. Forecasting Accuracy.....	39
	d. System Performance.....	42
	e. Micro Computer Suitability.....	48

B. AIR TO AIR.....	48
1. General.....	48
2. Lead for Target Motion.....	50
3. The Effect of Gravity Drop.....	56
4. Effect of Angle of Attack.....	56
5. Parallax Correction.....	57
6. Implementation and Timing.....	57
V. CONCLUSIONS.....	58
DISTRIBUTION OF FORECAST AIRSPEED ERRORS.....	60
RESULTS OF ACCURACY COMPARISON.....	63
STATIC ACCURACY SIMULATION PROGRAM.....	86
SIMULATION OF ANGLE OF ATTACK MODULE.....	87
FORECASTING SIMULATION.....	88
DELIVERY ACCURACY COMPARISON.....	94
PLM AIR TC GROUND PROGRAM.....	102
PLM AIR TC AIR PROGRAM.....	106
INITIAL DISTRIBUTION LIST.....	120

I. INTRODUCTION

The purpose of this paper is to examine the feasibility of using a micro computer in Naval aircraft to aid the pilot in the solution of the ballistics problem for unguided weapons. Specifically, this problem is to release a weapon at some point in space with the proper velocity vector so that the weapon will eventually strike a selected target. Since several models of attack aircraft (A-6E, A-7E) are equipped with computers which perform this function, special attention will be focused on those aircraft in which the pilot manually provides the solution. Such aircraft are referred to as fixed sight aircraft and the F-4 is representative of this class.

The micro computer, because of its low cost and weight, is a device with the potential to provide the pilot of fixed sight aircraft with useful and timely information. To determine whether such a potential could be exploited, the INTEL 8080 is examined as being representative of modern micro computers and the F-4 is considered as a typical fixed sight aircraft. The concepts, however, are intended to be more general in nature and no discussion, data, or conclusion contained herein is intended as an endorsement or criticism of the manufacturer or weapon system involved.

The body of the paper consists of four major sections. The sections are as follows: a description of the general nature of the unguided weapons delivery problem and how the problem is now solved by pilots of fixed sight aircraft; the proposed use of a micro computer to aid the pilot; an

analysis and test of the proposed solution; and conclusions.

II. UNGUIDED WEAPONS DELIVERY USING FIXED SIGHT AIRCRAFT

A. DEFINITIONS

Fixed sight aircraft is a term applied to those aircraft equipped with neither a lead computing gunsight nor a computer operated bomb release device. In such aircraft a sighting device or gunsight is located in the pilot's forward field of view. Its purpose is to represent visually an angle in milliradians (mils) measured in the positive direction below a longitudinal reference line called the FUSELAGE REFERENCE LINE (FRL). Figure 1 depicts such a device. Typically these devices are adjustable through a range of zero to approximately 200 mils, but once adjusted they remain set until manually changed by the pilot.

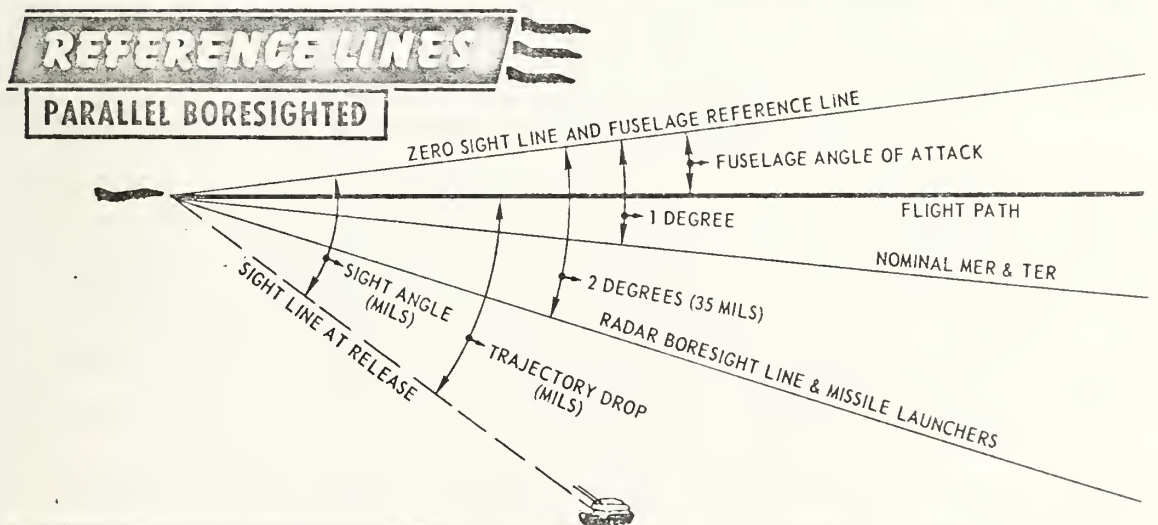
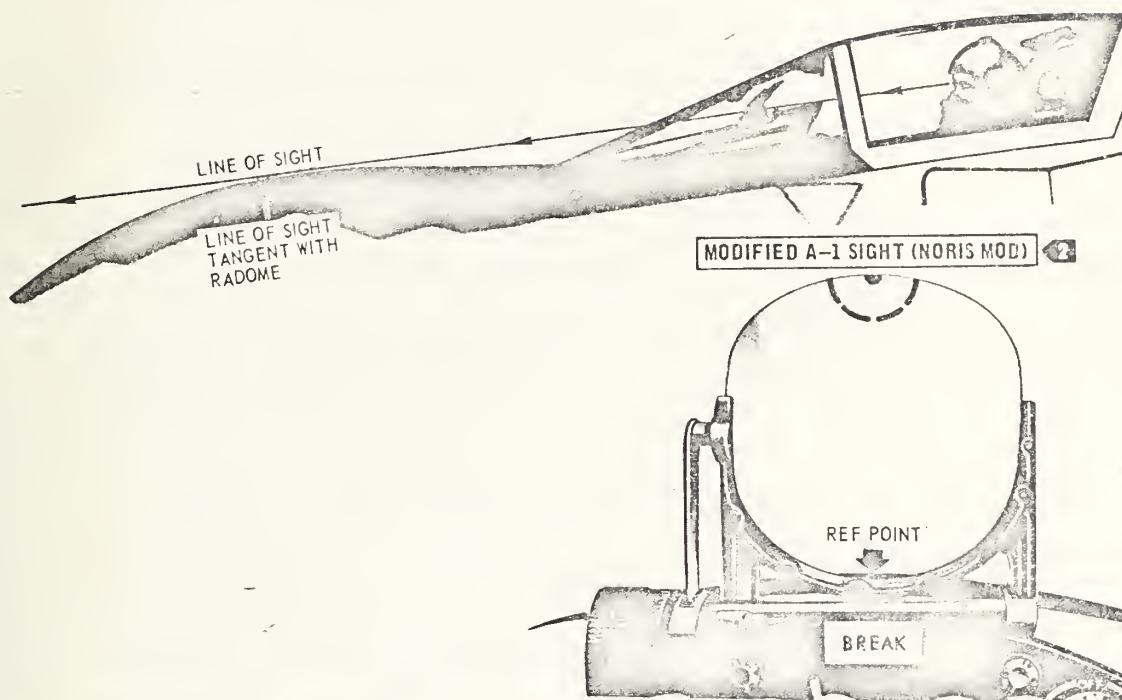


Fig 1 SIDE VIEW AND PILOT VIEW OF A GUNSIGHT

B• PROBLEM DEFINITION

For a particle of constant mass, m , moving under the action of an applied force, F , Newton's second law of motion states that

$$F = m \cdot a. \quad (1)$$

If the weapon is considered to be such a particle, and if the position of the weapon is measured in a rectangular coordinate system, (x, y, z) , then equation (1) may be written in vector form as

$$(a_x, a_y, a_z) = (F_x, F_y, F_z) / m. \quad (2)$$

For air to ground weapons delivery, the coordinate system is chosen so that y represents altitude, x represents the down range travel of the weapon, and the z axis measures displacement in azimuth. The origin is chosen so that, at the moment of release, x equals zero.

If it is assumed that the air mass through which the weapon is moving is stationary with respect to the earth, that gravitational forces are constant with respect to y , and that forces induced by the rotation of the earth are negligible, then $F_z = 0$. The remaining forces acting on the weapon are drag, thrust, and gravitational force. It will be assumed that thrust is zero, for all weapons except rockets. Rockets are assumed to have a short initial period of constant thrust after which they fall with the same discipline as all other unguided weapons. Drag acts in the

direction opposite to the direction of the velocity vector and is directly proportional to $||V||^2$ and air density. In a standard air mass, density is a decreasing function of altitude. Therefore equation (2) may be written in component form as follows

$$a_x = -\text{drag}_x * 1/m \quad (3)$$

$$a_y = -(\text{drag}_y + g*m)/m \quad (4)$$

where g is a constant 32.17 ft/sec^2 .

Letting $\text{THETA} = \text{ARCTAN}(v_y/v_x)$, (3) and (4) become

$$a_x = -\text{drag} * \cos(\text{THETA})/m \quad (5)$$

and

$$a_y = -\text{drag} * \sin(\text{THETA})/m - g. \quad (6)$$

Drag is represented as

$$\text{drag} = \rho * S * V^2 * C_d$$

where ' ρ ' is the air density, ' S ' is the cross sectional area, ' V ' is the total velocity $((V_x^2 + V_y^2)^{1/2})$, and C_d ,

the drag coefficient, is a dimensionless factor whose value depends on the shape of the weapon and the ratio of ' V ' to the speed of sound at altitude y . Equations (5) and (6) may be considered as differential equations

$$a_x = f(v_x, v_y, x, y, t) \quad (7) \text{ and}$$

$$a_y = h(v_x, v_y, x, y, t). \quad (8)$$

In air-to-ground weapons delivery, equations (7) and (8) may be solved numerically for any set of initial conditions

(v_x^0, v_y^0, x^0, y^0) to determine a time t' where $y_{t'}$ = the target

elevation and $x_{t'}$ is the down range travel of the weapon.

Therefore, the problem for air-to-ground delivery is to position the aircraft so that the initial conditions are matched when the selected aim point is a distance $x_{t'}$ down range from the aircraft.

For unguided air-to-air weapons, equations (7) and (8) may be solved together with initial value conditions to describe the position $(x(t), y(t), z(t))$ of the weapon at any time t after the weapon has been fired. Since the position of the target is also some function of time,

$(x_{\text{target}}(t), y_{\text{target}}(t), z_{\text{target}}(t))$, the problem is to find a set of initial value conditions such that for some time, t'' ,

$$(x(t''), y(t''), z(t'')) = (x_{\text{target}}(t''), y_{\text{target}}(t''), z_{\text{target}}(t'')).$$

In either case, the aircraft instruments provide the

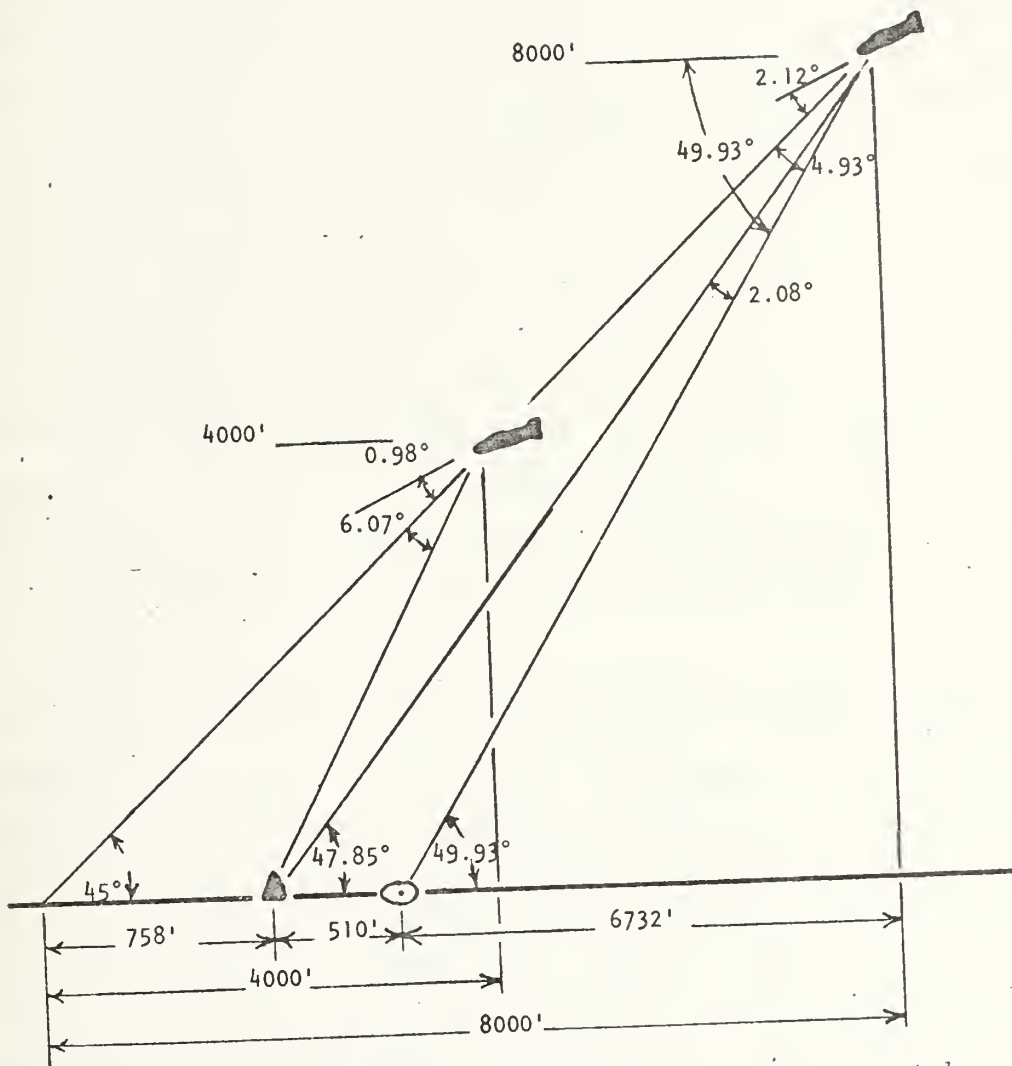
indication that the initial conditions are satisfied. The gunsight is used to establish a spatial relationship to the target. The desired terminal condition occurs at some future time when the weapon hits the target. This mixture of initial condition and terminal condition constraints makes the problem a combination initial value problem and boundary value problem.

C. PILOT SOLUTION FOR AIR TO GROUND DELIVERY

Although high speed computers which numerically solve (just prior to weapon release) equations (7) and (8) are installed on some attack aircraft (A-6E, A-7E), no such devices are installed on fixed sight aircraft. For the pilot of the fixed sight aircraft a numerical solution of the problem is not available in flight. Therefore sets of solutions have been calculated for various initial conditions. The pilot decides, prior to reaching the target, which of the precomputed sets of initial conditions he will match. Then he consults a table which contains a standard sight setting. After adjusting this setting to correct for variations in aircraft gross weight and release altitude, he adjusts his sight to this corrected value. Finally, he flies the aircraft so as to match the delivery parameters, places the sight image over the target, and releases the weapon. If all parameters have been matched, the weapon will have the greatest probability of striking the target.

In actual deliveries, however, the parameters are frequently not matched exactly. This is understandable considering that the controls available to the pilot (throttle and elevator control) do not directly, instantaneously or independently control the position or

initial velocity vector. Furthermore, dive angle information is not directly available from any aircraft instrument and the information which is directly available (true airspeed and altitude) must be mentally processed by the pilot before he can physically react, thereby adding further delay. Finally, the aircraft itself cannot react instantaneously to control inputs. Gauthier[1] has shown that sudden perturbations of the angle of attack caused by pilot induced changes in pitch attitude (elevator control) will cause the information provided by a fixed gunsight to be unreliable for several seconds. In addition, as shown in Fig 2, the gunsight image is constantly moving across the earth so that the proper sight 'picture' occurs precisely once during the dive.

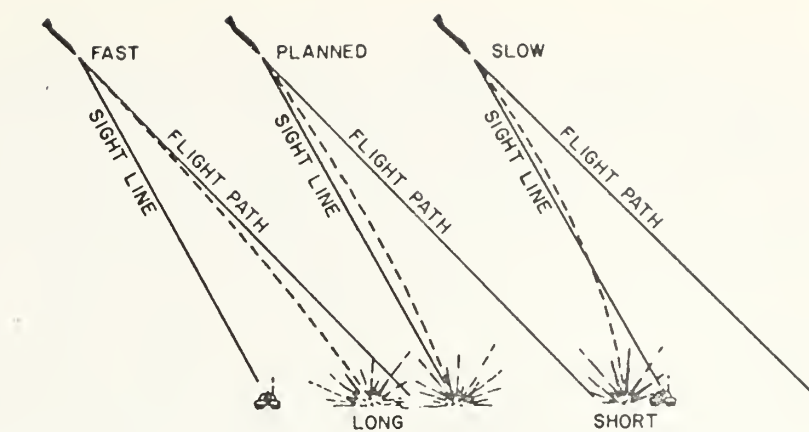


NOTE: Although Depression from flight path (106 mils) extracted from the ballistic tables is corrected for parallax, the remaining calculations ignore these factors. Therefore it is apparent that there are small geometric anomalies in the above diagram.

Fig 2 SIGHT LINE MOTION IN A DIVE

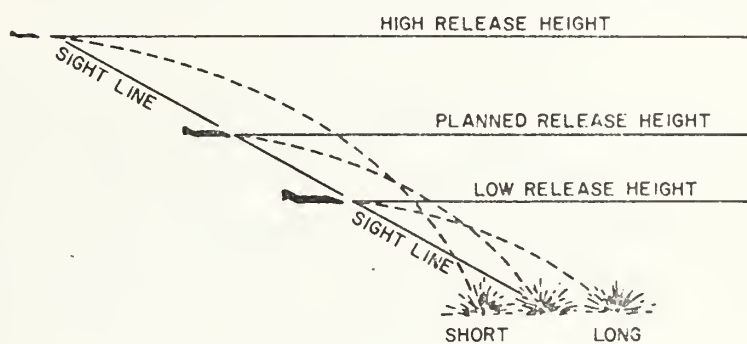
In spite of these difficulties, there are strategies which help the pilot to learn a solution to the problem. First, the deliveries have been standardized so that the pilot may concentrate his efforts at mastering several basic maneuvers. This practice tends to reduce the magnitude of delivery errors. The second aspect which aids the pilot is known as error analysis. Since actual conditions rarely match those of the practice patterns, the pilot is taught to recognize those errors which are present and to apply corrective measures. Fig 3 graphically depicts the effect of various errors in airspeed, dive angle, release altitude, and aim point. For each of the standard deliveries, there is a table in reference [2] which lists, for each weapon, the effect upon weapon impact of errors of 10 knots in true airspeed (fast and slow), one degree in dive angle (steep and shallow), 100 feet in release altitude(high and low), and one milliradian in sight setting (plus and minus).

To use the tables, the pilot memorizes key values for the weapons he is going to release. During the tracking portion of his dive, he first attempts to minimize the errors in each of the delivery parameters. If he decides that the value of some or all of the delivery parameters will not equal the preselected values, he estimates the cumulative effect of all such errors on weapon impact. Having estimated this miss distance, he then deliberately causes another error either in release altitude or sight angle to counteract the effect of the previous errors.



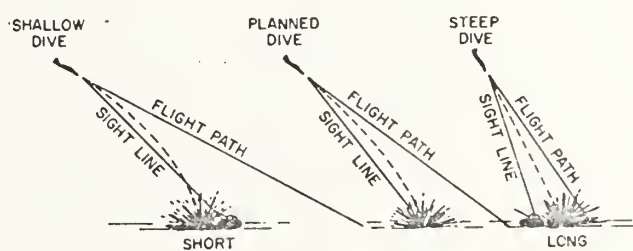
RELEASE HEIGHT ERROR

LEVEL DELIVERY



DIVE ANGLE ERROR

CONSTANT RELEASE HEIGHT AND AIRSPEED



SIGHT ANGLE ERROR

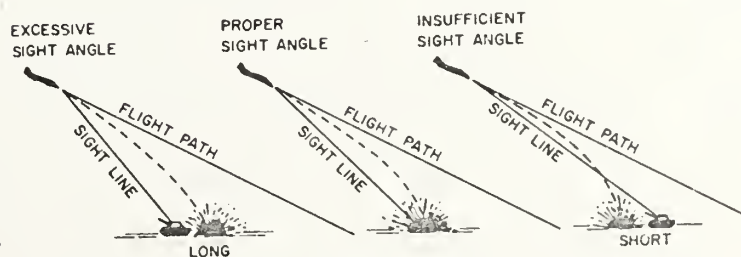


Fig 3 EFFECTS OF AIRSPEED, HEIGHT, DIVE, AND SIGHT ERRORS

As an illustration of this method, suppose the pilot estimates that he will be 10 knots fast and one degree steep at release altitude and that the combined effect of the errors will place the weapon 100 feet past the target. If, in the error sensitivities chart for the particular weapon, the entries for +100 ft. and -1 mil are -25 feet and -10 feet respectively, then the pilot may choose to release the weapon 400 feet high in relation to the preselected release altitude or he may use an imaginary sight line which represents an angle of 10 mils less than the preselected angle.

In view of the difficulties involved in fixed sight dive bombing, it is not surprising that considerable effort has been devoted to the development and implementation of computer systems to improve the delivery accuracy of all current generation Naval Attack aircraft.

D. AIR TO AIR GUNNERY

In air-to-air gunnery, the pilot of a fixed sight aircraft essentially solves his aiming problem in the same manner as for air to ground weapons delivery. He selects several representative situations and consults a table to determine the correct sight setting for each particular situation. In air to air gunnery, however, the pilot would usually like to be able to fire his aircraft's guns at any time he has a good chance of hitting the target. Therefore, rather than try to match any preselected set of firing conditions, the pilot takes the best guess and fires. The guns are loaded with a mixture of tracer rounds so that the pilot visually adjusts his aim depending on his evaluation of the effect of previous aim angles.

III. PROPOSED SOLUTION

In order to achieve the full benefit of the low cost of the micro computer, it was assumed that no additional sensor equipment such as doppler, air-to-ground ranging devices, or inertial navigation devices would be included in the modified system. Instead, the system would be designed around existing components with ultimate responsibility for problem solution still resting with the pilot. The micro computer is used chiefly to perform those calculations which the pilot of fixed sight aircraft now estimates.

A. COMPONENTS

The components of the modified system are depicted in Fig 4. Their relationship to the overall system is given below.

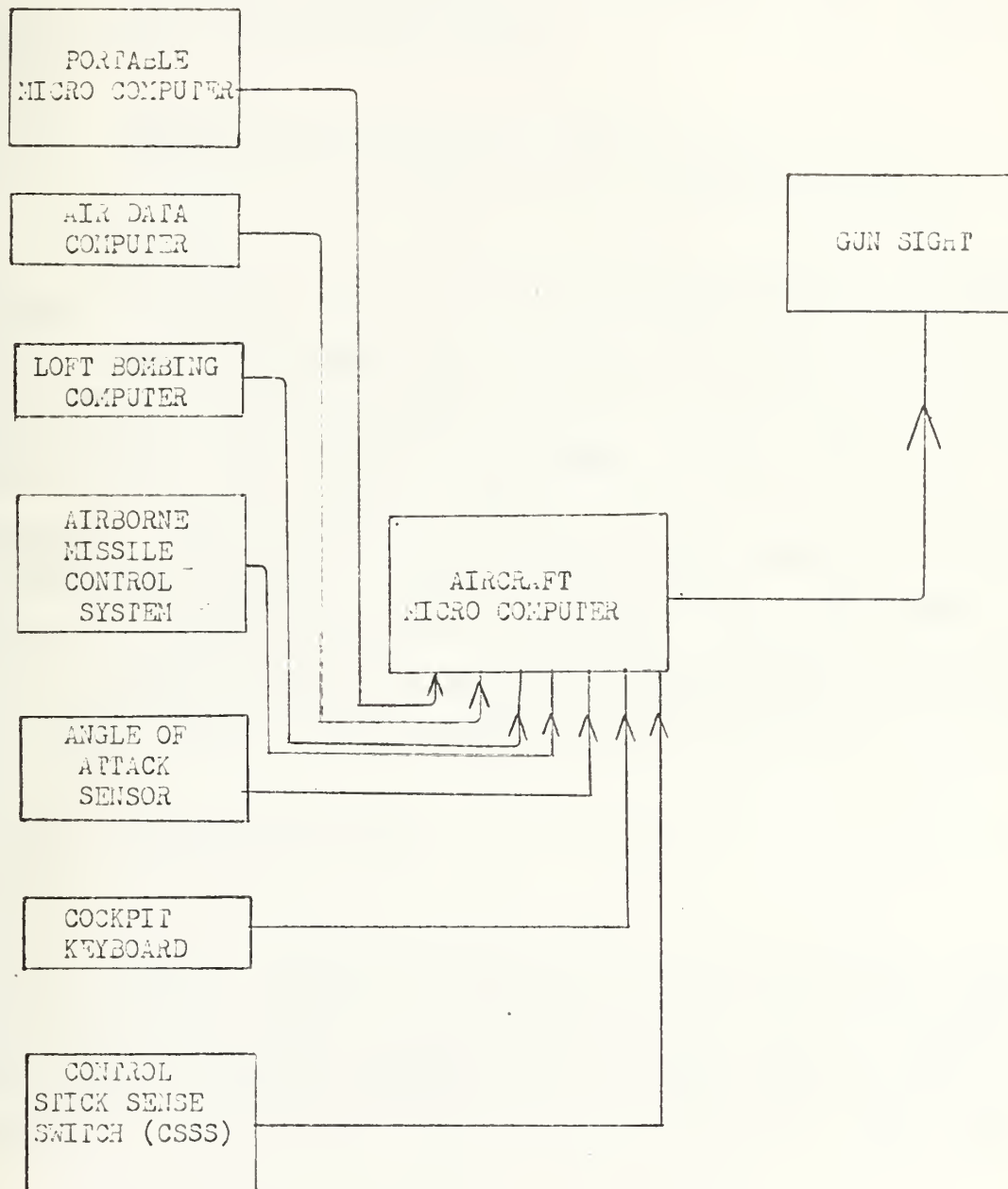


Fig 4 SYSTEM COMPONENTS

1. Control Stick Sense Switch

An on/off switch which is activated 'on' when the pilot begins tracking the target.

2. Portable Micro Computer With Auxiliary Memory

A device whose auxiliary storage contains a complete library of standard release parameters and error sensitivity factors for each weapon authorized for carriage and release from the particular aircraft. Each weapon would require approximately 128 bytes of storage. It is assumed that no more than 50 such weapons would apply to any particular aircraft. The ground based micro computer would be capable of retrieving blocks of such data and passing the data to the micro computer aboard the aircraft. It is also assumed that no more than three different weapons would be carried by a particular aircraft on any flight.

3. Air Data Computer

A device currently installed on Naval aircraft which provides analog signals corresponding to true airspeed, calibrated airspeed, and altitude. These signals would be converted to digital format to be read by the micro computer.

4. Loft Bombing Computer

An attitude reference gyro of the type currently installed in the F-4 whose analog signals corresponding to

pitch and roll angle as well as rate of pitch and heading change would be converted to digital format to be read by the micro computer.

5. Cockpit Keyboard

Any standard numerical keyboard, of the type used on pocket calculators, which has at least five function switches and light emitting diodes to confirm register contents. This device would be used by the pilot, or preferably by a crewman, to input the mode (air-to-air or air-to-ground), and if in the air-to-ground mode to input the desired delivery, weapon to be released, target elevation, and aircraft gross weight.

6. Micro Computer

A micro computer with a word size of at least eight bits and speed at least that of the INTEL 8080 (2.5 micro second instruction cycle).

7. Gunsight

Either a standard fixed gunsight modified by replacing the fixed reticle light source with an array of addressable light emitting diodes, or a standard 'heads up' gunsight.

8. Airborne Missile Control System

One of the several models of air-to-air radars currently installed on the F-4 which provides analog signals

corresponding to the present range to the target and the rate of closure. These analog signals would be converted to digital format to be read by the micro computer.

9. Angle of Attack Sensor

A device which is currently installed on Navy jet aircraft which provides an analog signal corresponding to the angle between the flight path of the aircraft and the FUSELAGE REFERENCE LINE. This signal would be converted to digital format for use in air-to-air calculations.

B. DESCRIPTION OF USE

Fig 5 shows the overall timing sequence for the functions of the system.

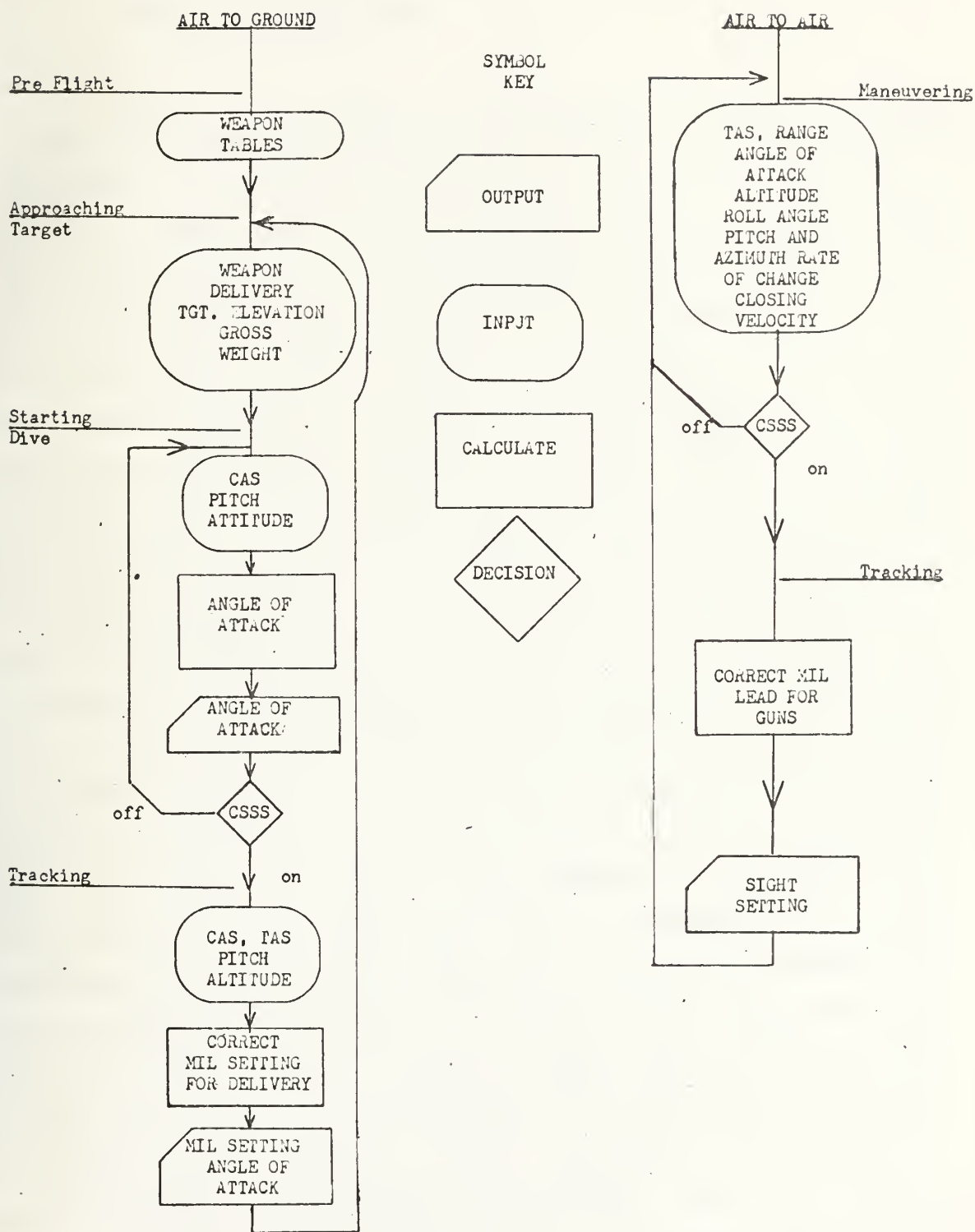


Fig 5 TIME DIAGRAM

1. Air to Ground Mode

Prior to flight, the aircraft is loaded with some authorized ordnance configuration. After the aircraft is operating on internal electrical power, a portable micro computer loads the appropriate tables into the memory of the aircraft's computer. As the aircraft approaches the target, the pilot/crewman keys a standard delivery and weapon code into the micro computer along with the target elevation and aircraft gross weight. As the pilot commences his roll-in, the micro computer samples calibrated airspeed and pitch angle and calculates the angle of attack of the fuselage reference line for a steady state dive at that pitch angle and calibrated airspeed and displays the angle of attack in mils on the gunsight. The pilot uses this information to establish the aircraft in a dive path which intersects the earth at some predetermined distance beyond the target (aim off point). When the aircraft is established in the proper dive, the pilot actuates the control stick sense switch and the computer begins to sample true airspeed, calibrated airspeed, altitude, and pitch angle. The pilot makes no further throttle adjustments. The computer calculates dive angle and estimates what the release parameters (airspeed and dive angle) will be when the aircraft reaches the prescribed release height above the earth. This sampling and calculation takes about two seconds. The computer then calculates the effect of any errors in the estimated airspeed and dive angle and corrects for the impact error by addressing a new diode in the gunsight or sending a correction signal to the heads up gunsight. The new sight setting would be calculated using the error sensitivity factors which had been read prior to flight. The pilot uses the corrected sight setting as the new aiming reference and places the revised sight pipper over the target at release

altitude and manually releases the weapon. After dive recovery, the computer is reset and the process is repeated as desired.

2. Air to Air Mode

For air to air use, there would be no need to load tables into the aircraft computer prior to flight. Instead, the program and appropriate tables would remain in the computer memory.

In flight the pilot/crewman would select the air-to-air mode. The computer would then begin to sample true airspeed, altitude, present range to the target, closure rate, roll angle and pitch and azimuth rates of change.

At this time the 35 mil setting would be illuminated on the gunsight. When the pilot had established a suitable tracking position with the sight held steady on the target, he would activate the control stick sense switch.

The computer would then calculate the proper lead angle for the most recently sampled conditions and display this lead angle on the gunsight. The pilot would adjust to the new aim angle as smoothly and quickly as possible and fire the guns. Tracers would still be mixed with other ammunition so that the pilot could adjust his fire as necessary. The time required to perform the calculations would be about .2 seconds.

IV. ANALYSIS AND TEST

A. AIR TC GROUND

1. General Considerations

If it is assumed that density, temperature, and the speed of sound are continuous, differentiable functions of altitude, then there exist unique solutions of equations (7) and (8) for each set of initial conditions. Moreover, it is intuitively clear that if the solutions were written in the component form

$$x(t) = F(v_x^0, v_y^0, x^0, y^0, t) \quad (9)$$

and

$$y(t) = H(v_x^0, v_y^0, x^0, y^0, t), \quad (10)$$

then the position of the weapon at time t could be considered to be a continuous function of the initial conditions. That is, for any neighborhood, N , of a point $(x(t), y(t))$ which represents the position of the weapon at time, t , after release (with initial conditions

(v_x^0, v_y^0, x^0, y^0)), there exists a neighborhood, U , of the initial conditions such that any identical weapon released with initial conditions which were in U would, at time, t , be in N . Furthermore, it may be assumed that first partial derivatives of F and H exist within U so that

$$dX = F_{v_x} * dv_x^0 + F_{v_y} * dv_y^0 + F_x * dx^0 + F_y * dy^0 \quad (11)$$

and

$$dY = H_{v_x} * dv_x^0 + H_{v_y} * dv_y^0 + H_x * dx^0 + H_y * dy^0. \quad (12)$$

Where the variables and partials on the right represent changes in the initial conditions and those on the left represent position at some time, t , which is held constant.

Since the proposed system incorporates no additional sensor devices, it must perform essentially the same type of calculations which the pilot of a fixed sight aircraft now performs. It samples airspeed, altitude, and pitch attitude and predicts what the value of true airspeed and dive angle will be at release altitude. These values are then compared with the standard release parameters for the particular delivery being used. Any difference between the predicted value and the desired value is converted to a gunsight correction which must be applied to the sight to counteract the effect of the release airspeed or dive angle error. The sum of the correction for airspeed and the correction for dive angle is then applied to the gunsight and the pilot flies the aircraft to place the new sight image over the target at release altitude.

Since the proposed solution relies on the error sensitivity tables, and since these tables are based on the

partial derivative of F with respect to several of the components of the initial condition (for example, an error in true airspeed changes v_x , v_y , and x), and since the solution assumes that effects of changes in airspeed, dive angle, and aim angle are linear and independent, an analytical proof of the validity of these assumptions would require an explicit representation of F and H . Unfortunately, there is no known closed form solution available to cover all the possible sets of initial conditions. Therefore, a series of computer experiments were performed to determine whether the method of error sensitivity analysis was suitable.

2. Tests of the Proposed Solution

a. Static Accuracy

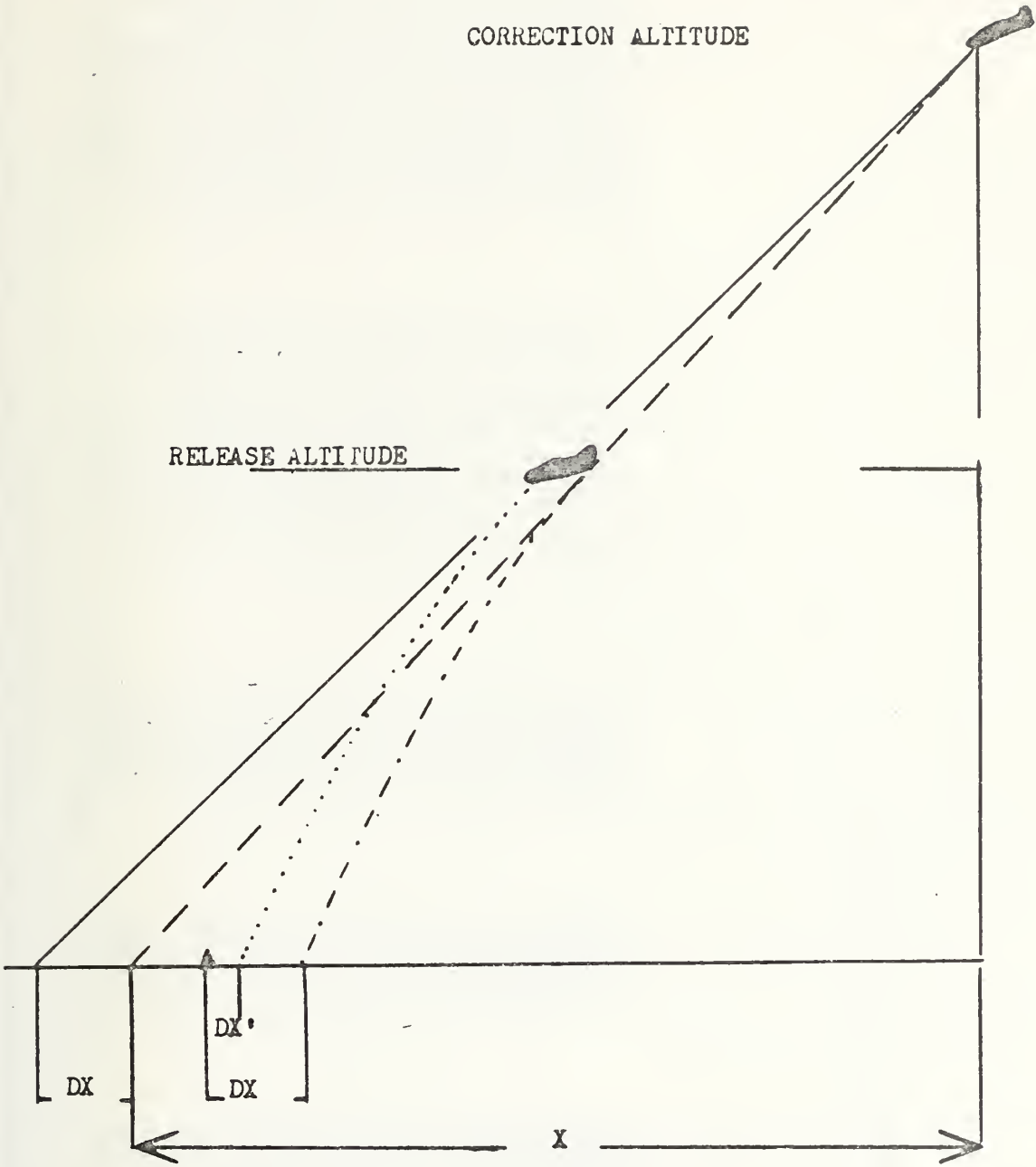
The first test of the proposed solution was designed to measure the effectiveness of the error analysis technique. To separate the error analysis technique from any errors which might be induced by airspeed prediction or dive angle calculation, it was assumed that these functions were performed exactly by the micro computer. To measure the ability of the system to improve accuracy, an arbitrary value of plus 200 feet or minus 200 feet was assigned to the uncorrected impact error, DX in figure 6. The original dive angle was assigned a value which was five degrees steep or five degrees shallow. It was assumed that the correction which was applied to the gunsight would cause the bomb to hit the target if the pilot could establish a parallel dive path whose displacement from the original dive path was $-DX$ feet (measured along the X axis). It was assumed that, instead of performing such a maneuver, the pilot used the

new sight image to make a single change in his dive path. This change occurred three seconds prior to reaching release altitude and moved the aim off point $-DX$ feet. This maneuver causes a small change in the dive angle for which the gunsight was not compensated. At release altitude, the effect of this change in dive angle was computed using the error sensitivity tables for the Mk-76 Practice Bomb (DX' figure 6).

A FORTRAN program was executed to determine the values of DX' , based on the previous assumptions. The program is listed in Appendix C and results are in Table 1. In general the release error sensitivity method improved accuracy by a factor of eight, and, in the worst case, by a factor of five.

CORRECTION ALTITUDE

RELEASE ALTITUDE



- DIVE BEFORE CORRECTING
- DIVE AFTER CORRECTING
- - . - - . UNCORRECTED BOMB
- CORRECTED BOMB

Fig 6 SIGHT ANGLE CORRECTION IN A DIVE

b. Real Time Accuracy

An important part of the proposed solution was the calculation of the angle of attack of the fuselage reference line. The pilot would use this information to establish the aircraft in a straight path which intersected the ground at a precomputed distance beyond the target, called the aim off point, reference [3]. The computer would also use this angle of attack information to compute the actual dive angle of the aircraft.

Although all Naval carrier based jet aircraft are equipped with angle of attack sensors, these devices are designed primarily to provide accurate information in the landing phase of flight and are inherently inaccurate during high speed low angle of attack situations, reference [4].

Therefore, in order to have a more accurate value for the angle of attack, figure 2-56 of reference [2] was used as a source, . Fig 7.

FUSELAGE ANGLE OF ATTACK

DATA BASIS: FLIGHT TEST

HEAD STANDARD DATA	
ALT. FT.	TEMP. °C
5,000	15.0
10,000	5.1
15,000	-4.8
20,000	-14.7
25,000	-24.6
30,000	-34.5
35,000	-44.4
40,000	-54.3
45,000	-56.5
50,000	-56.5

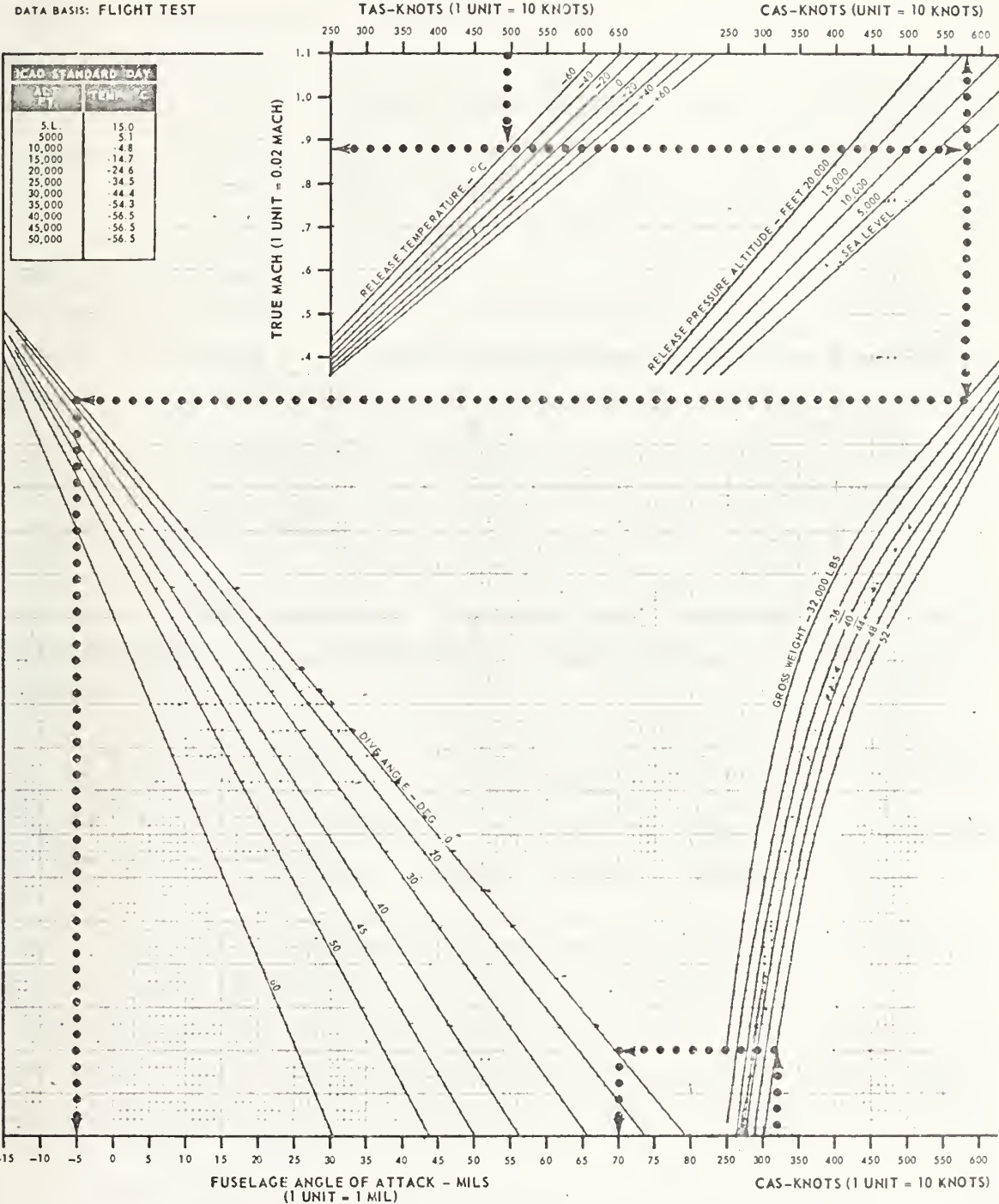


Fig 7 ANGLE OF ATTACK CHART

Figure 7 is the chart which a pilot would use to calculate the angle of attack for any particular air to ground delivery condition. The chart may be used with either of two sets of input keys. The first set is dive angle, temperature, true airspeed, and gross weight. The second set consists of calibrated airspeed, gross weight, and dive angle. Since the central air data computer provides calibrated airspeed information, the second set was used. To calculate manually the angle of attack for a given calibrated airspeed, gross weight, and dive angle the following procedure is used:

- (1) Find the desired calibrated airspeed (cas) on the right half of the bottom of the chart.

- (2) Move vertically to the curve labeled with the appropriate weight.

- (3) Proceed horizontally to the left until intersecting the straight line labeled by the desired dive angle.

- (4) Proceed directly down the chart and read the desired value of the angle of attack in mils from the bottom of the chart.

To reproduce the information contained in figure 7, it was necessary to represent the weight curves and straight lines in an algebraic form. The form chosen was the polynomial. Since all air-to-ground charts are based on the assumption that the aircraft gross weight is 42,000 pounds with scalar adjustments for different weights, the same scheme was adopted for the micro computer system. The 42,000 pound curve was approximated with a second degree interpolating polynomial, $F(cas)$. The straight lines were

considered to be of the form

$$y = m * x + b$$

where 'y' represents ordinate values, 'x' represents abscissa values, 'm' is the slope of the line, and 'b' is the 'y' intercept value. Substituting ALPHA, the angle of attack, for 'x', F(cas) for 'y', m(THETA) for m, and B(THETA) for 'b', the following equation was evaluated:

$$ALPHA = (F(cas) - B(THETA)) / m(THETA).$$

F was a quadratic interpolation of the 42,000 pound curve. The argument of F, cas, was taken as the difference between calibrated airspeed and 295. This linear translation allowed single byte multiplication to be used. THETA was the dive angle. B was a tabled, first divided difference function of the zero mil intercept values of the straight lines in the lower left quadrant of figure 7. M, the slope of these straight lines, was found to approximate closely the value of '-SECANT(THETA)'.

Therefore, in order to speed the calculations, the equation which was actually solved was

$$mil = \cos(THETA) * (B(THETA) - F(cas)). \quad (13)$$

The COSINE calculation was based on a divided difference table which was arranged in four degree increments.

An ALGOL program was executed to simulate an INTEL 8080 fixed point arithmetic evaluation of the preceding equation. A listing of the program is contained in Appendix C. The results of the ALGOL simulation appear in table 2. A FLM version of the routine was timed by the INTERP8 as requiring 4798 micro seconds.

AIRSPEED CAS												
	300	320	340	360	380	400	420	440	460	480	500	520
DIVE DEGREE												
10	61	51	43	36	31	26	21	18	16	15	15	16
20	59	49	42	35	30	25	20	17	15	15	15	15
30	51	43	36	30	26	21	17	15	13	13	13	13
40	48	40	34	29	24	20	16	14	13	12	12	13
50	34	29	24	20	17	14	11	10	9	8	8	9
60	31	26	22	18	15	13	10	9	8	7	7	8

TABLE 2. COMPUTED FUSELAGE REFERENCE LINE ANGLE OF ATTACK

c. Forecasting Accuracy

A third topic for investigation is the ability of the micro computer to forecast accurately the value of the release true airspeed early in the dive. The basic forecasting scheme is to fit a parabola through a series of samples of airspeed and altitude. It is assumed that both values can be read to the equivalent of four decimal places of precision. In fitting the parabola, altitude is treated as the independent variable and the release true airspeed is assumed to be

$$\text{airspeed} = C_1 + C_2 * \text{relalt} + C_3 * \text{relalt}^2 \quad (15)$$

where 'relalt' is the calculated release altitude.

To determine whether such a method was appropriate, it was necessary to first construct a model of the F-4 in a dive, and then write a computer program to simulate the model. The model chosen was that of Newton's second law of motion:

$$F = m * a$$

where force, F, was assumed to be

$$\text{Thrust} + W * \sin(\text{THETA}) - \text{drag}.$$

Thrust and drag were opposing forces acting parallel to the flight path, THETA was constant, and thrust, mass, and weight, W, were also constant.

The resulting differential equations were

$$A = (\text{Thrust} + W * \sin(\text{THETA}) - \text{drag}) / \text{mass} \quad (16)$$

and

$$V=A*t. \quad (17)$$

Drag was assumed to be a function of dive angle and true airspeed. Figure 2-61 of reference [2] was selected as the source of data from which to obtain values of drag. This particular figure represents true airspeed vs. altitude at various dive angles for a particular engine speed, gross weight, and external configuration (drag index 60). The relationship is represented by a series of smooth curves, where each curve corresponds to a particular dive angle. The curves themselves were first represented using an interpolating second degree polynomial determined by a least squares fit of eight evenly spaced sets of data points. The coefficients of the polynomial were calculated by the Naval Postgraduate School Computer Library subroutine LSQPL2. With these coefficients, it was then possible to determine for any altitude and dive angle a true airspeed value based on the curves.

Using the relationship previously established, it was possible to determine an approximate value for the time required to fly from one altitude to another. Using increments of 10 knots and assuming that the rate of change of velocity was linear within the interval, an empirical approximation for the acceleration in a particular time increment was calculated. The value of the drag during that time increment could then be calculated. This method was used to establish pairs of data points (true airspeed, drag) for each of the dive angles given. An additional pair (0,0) was added for each dive and the resulting data was again fitted using the routine LSQPL2. The coefficients were used in FORTRAN subroutine DRAG, see Appendix E.

With the drag function computed, equations (16) and (17) were numerically solved using a standard

Runge-Kutta fourth order integration scheme. this was done in a FORTRAN subroutine, INTEG, (see Appendix E).

Using the model, three forecasting schemes were tested. The first scheme consisted of sampling airspeed and altitude values at one second intervals. A Newton divided difference interpolating polynomial was then generated and release true airspeed was predicted as

$$\begin{aligned} \text{tas}_{\text{release}} = & \text{tas}_{t_0} + (\text{release alt} - \text{alt}_{t_0}) * (F_{(0,1)} \\ & + (\text{release alt} - \text{alt}_{t_1}) * F_{(0,1,2)}). \end{aligned} \quad (18)$$

$$F_{(0,1)} = (F_0 - F_1) / (x_0 - x_1),$$

$F_{(1,2)}$ is analagously defined,

$$F_{(0,1,2)} = (F_{(0,1)} - F_{(1,2)}) / (x_0 - x_2),$$

X represents the independent variable and F represents the dependent variable.

The second method consisted of taking a total of twelve samples during a two second period. The first four samples were added and the average was taken as the true value. Similar averages were used to represent the two remaining inputs to a Newton divided difference second degree interpolating polynomial of the type described above.

The third prediction scheme was to take 20 samples during the same interval and to generate a least squares orthogonal polynomial to fit the data. The polynomial was also a parabola and it was generated and evaluated based on subroutines contained in reference [5].

For the experiment, 1000 dive runs were simulated. Each run was based on the following assumptions:

(1) Entry airspeed equals 345 knots.

(2) Thrust equals 20,000 pounds.

(3) Dive angle equals 30^0 .

(4) Entry altitude equals 6000 feet.

(5) True airspeed and altitude would be read to four decimal digits of precision.

(6) On any particular run, the altitude and airspeed error would remain constant. Between runs the error would be normally distributed with the mean airspeed error of zero and standard deviation of .5 knot. The altitude error would have a mean of zero and a standard deviation of 10 feet.

The latter distributions are the result of assuming that the errors were normally distributed with the maximum allowable error equal to three times the standard deviation. Allowable instrument errors were obtained from reference [6].

Normal distributions were simulated using the Naval Postgraduate School Computer Library package LLRANDOM. The test was repeated with a different seed to verify original results. See Appendicies A and E.

d. System Performance

To determine whether the use of the micro computer would significantly improve the delivery accuracy of the F-4 in the air to ground environment, an experiment was conducted based on the following assumptions:

(1) The previously described model represented an F-4 in a dive bombing run.

(2) The Mk-81 Low Drag Bomb is a typical unguided air to ground weapon.

(3) In the simulation, it would be assumed that the pilot, at release altitude, would have the sight aimed at the target.

(4) All weapons would be released from a position 7 feet below and 18 feet aft of the gunsight.

(5) All deliveries would be in no-wind conditions.

(6) The path of the Mk-81 Low Drag Bomb could be described by the following set of equations which were obtained from reference [7]:

$$X'' = -E * X' \quad (19)$$

and

$$Y'' = -E * Y' - g \quad (20)$$

where X represents down range travel, Y represents altitude, X', and Y' are the horizontal and vertical components of the velocity vector. The value of the other variables was determined as follows:

$$E = (c * p * V_a^2 * K_d) / 144$$

where

c is the reciprocal ballistic coefficient
(diameter² / weight),

p is the air density $0.0751366 * e^{-(0.00003158 * Y)}$,

V_a is the true airspeed in ft/sec².,

and

K_d is the drag coefficient.

The experiment consisted of simulating 1000 dive runs for each of the following standard deliveries:

20⁰ dive, 2000 ft. release, 450 knots true airspeed

30⁰ dive, 3000 ft. release, 450 knot true airspeed

30⁰ dive, 5000 ft. release, 500 knot true airspeed

45⁰ dive, 4000 ft. release, 450 knot true airspeed

45⁰ dive, 6000 ft. release, 500 knot true airspeed

60⁰ dive, 7000 ft. release, 500 knot true airspeed.

On each run the sampling method previously described as one second interval sampling was used. The predicted airspeed was calculated by equation (18). The entry conditions were considered to be normally distributed

about a correct set of (airspeed, altitude, thrust, and dive angle) which was calculated to achieve the desired delivery parameters. The standard deviations of the distributions of the entry conditions were 10 knots for true airspeed, 200 feet for altitude, 2.5^0 for dive angle, and 750 pounds for thrust.

Normal distributions were simulated using the Naval Postgraduate School subroutine LLRANDOM. The standard deviations were assigned with a data statement. Two pseudorandom variables, x and y , were obtained from the LLRANDOM package. The variable x was obtained from a uniform $(0,1)$ distribution and y was obtained from a normal $(0,1)$ distribution. The desired value, z , was then calculated as

$z = \text{DMEAN} + y * \text{STDV} * \text{COIN}(x)$ where DMEAN is the desired mean, STDV is the standard deviation, and COIN is a statement function which returns a +1 if $x < .5$ and a -1 otherwise.

After the release conditions had been predicted by the subroutine SAMP, two conditions were simulated. First, that the aircraft had no micro computer and that the pilot continued in a steady dive to release altitude. Second, that the pilot had a micro computer, that the micro computer supplied his gunsight with a corrected sight setting equal to the value calculated by SAMP, and that the pilot adjusted his aircraft's dive path as soon as this information was available, so that at release altitude, the new sight setting was 'on' the target.

For each condition, a simulated bomb was dropped at release altitude. A miss for the bomb was calculated as the distance between the point where the sight angle

intersected the earth and the down range travel, x , which was numerically computed (based on the equations (19) and (20)). In addition, in the case of the modified flight path, a correct sight setting was also calculated. Finally, the ratio of the miss distance with the micro computer to the miss distance without the micro computer was calculated. If the micro computer gave a greater miss than the basic aircraft, the ratio was set to plus or minus 1.1, depending on whether the errors were both on the same side of the target.

Arrays were built with these 1000 values and histograms were printed showing the distribution of the various values. See appendix B. The FORTRAN routine which simulated the experiment is in appendix F. An additional table was constructed from the histograms which compares the median miss distance under the assumptions that the micro computer was or was not used. See table 3.

RELEASE PARAMETERS			RANGE ERROR PROBABLE *	
DIVE	ALTITUDE	AIRSPEED	UNCORRECTED	CORRECTED
20	2000	450	88	44
30	3000	450	68	16
30	5000	500	120	20
45	4000	450	70	28
45	6000	500	139	53
60	7000	500	60	32

* Range error probable is the distance from the target of the median bomb.

TABLE 3. Mk 81 BOMB DROP SIMULATION

e. Micro Computer Suitability

The previously tested error sensitivity algorithm was converted from FORTRAN to PLM for test on the INTEL 8080. The program was compiled on the Naval Postgraduate School IBM 360/65 and interpreted using the INTERP8 subroutine. Timing data was obtained and found to be .75 seconds for the entire program. See Appendix G. The program was then executed on the INTEL 8080 and final output (sight setting) was found to be in agreement with sample results which had been obtained from executing the FORTRAN version on an SDS 9300 computer.

B. AIR TO AIR

1. General

In air-to-air gunnery considerations, it must be assumed that the position of the target is some function of time,

$$(x,y,z) = F(a(t),b(t),c(t)).$$

Since, in general, no assumptions can be made to restrict this motion relative to the (x,y,z) coordinate system of air to ground considerations, a new coordinate system is introduced. The origin of the new system, (x',y',z') is the muzzle of the fighter aircraft's cannon. The x' axis is always oriented so that a bullet's initial velocity vector will be aligned with the positive x' axis. The y' axis is perpendicular to the x' axis and forms, with

the x' axis, a plane which contains the wing roots of the fighter. The z' axis is orthogonal to the (x', y') plane with the positive direction measured in the same direction as the vertical tail of the fighter.

It should be noted that the x' axis is not always aligned with the gun barrel boresight line. This is caused by the bullet's velocity vector being the vector sum of the basic muzzle velocity vector and the aircraft velocity vector. When the aircraft is flying at an angle of attack greater than zero these two vectors are not coincident. Therefore, the x' axis is shifted away from the gun boresight line toward the relative wind.

The set of points from which a fighter can effectively employ guns against an airborne target lies within a 30 degree cone emanating from the tail of the target. The minimum firing range within this cone is approximately 500 feet and the maximum range is approximately 2000 feet.

The dimensions of this firing region effectively restrict the problem so that, for a bullet to hit the target, the time of flight is less than two seconds. This short time of flight, together with the basic muzzle velocity of the standard aircraft cannon (3300 fps), allow the computations of the proper lead angle to be performed in four distinct steps, Reference [8]. The steps are as follows:

- (1) Estimate the future angular displacement of the target from the present line of sight. This displacement is assumed to be caused by the target's present velocity vector acting as a constant during the time of flight of the bullet. This angle is called the lead for target motion.

(2) Calculate the effect of gravity drop.

(3) Calculate the effect of the fighter's angle of attack.

(4) Calculate the effect of the distance between the gun and the gunsight (parallax).

2. Lead for Target Motion

The calculation of the lead for target motion in a plane parallel to the (y', z') plane is based on the following analysis of the situation:

(1) If the fighter is tracking the target, that is, the fighter is in the firing cone with the 35 mil pipper held steady on the target, the relative positions of the two aircraft are 'close' to the positions needed for successful gun firing.

(2) Therefore, assume that the sight line and the x' axis are coincident.

(3) At this time, the fighter's velocity minus the rate of closure represents the target's velocity component parallel to the x' axis.

(4) The rate at which the fighter is turning, in radians/second, multiplied by the range is a close approximation to the target's velocity component parallel to the (y', z') plane.

(5) If there exists a time, t' , such that a bullet fired at time $t=0$ will strike the target at time t' ,

then the geometry of the flight of the bullet in a plane containing the x' axis and the component of the velocity vector of the target which is in the plane parallel to the (y',z') plane, can be approximated by Fig 8.

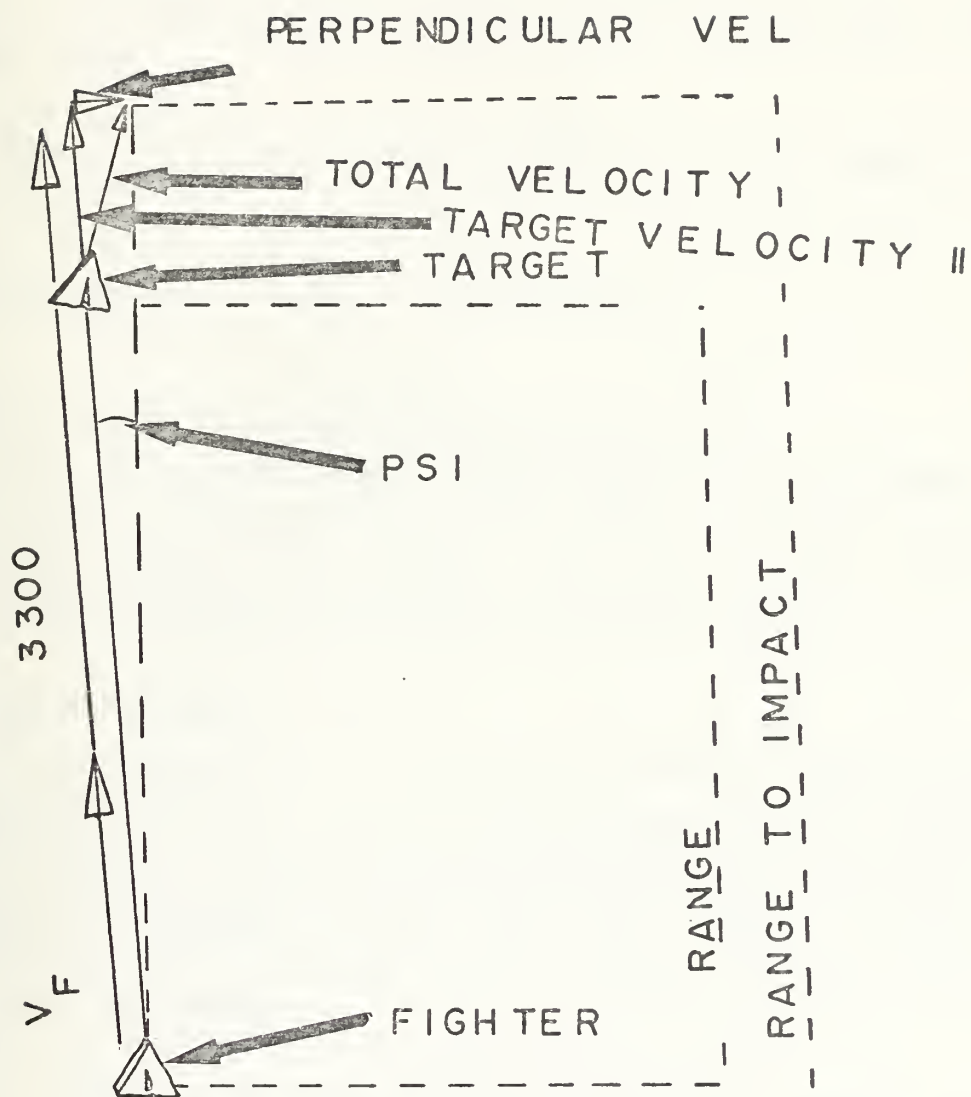


Fig 8 AIR TO AIR GUNNERY DIAGRAM

(6) Therefore, the lead for target motion orthogonal to the x' axis can be approximated by the angle, ψ , (see Fig 8).

The calculation of the angle in mils uses the fact that

$\lim_{x \rightarrow 0} \sin(x)/x = 1$. This implies that, for small angles,

$$\text{ARCSIN}(x/y) \doteq x/y \doteq \text{ARCTAN}(x/y)$$

where \doteq means "is approximately equal to".

In fact, for angles smaller than 200 mils, the maximum error of these approximations is 4 mils. Since the set of possible answers lies within the range 0-200 mils, the following equation was solved to determine ψ .

$$\psi = V_0/V_{x'}$$

V_0 , the component of the target's velocity vector which was orthogonal to the x' axis, was calculated as the product of the fighter's rate of turn multiplied by the range to the target, multiplied by 1000. $V_{x'}$, the average velocity of the bullet, was calculated as the range to impact divided by the time of flight.

Since the calculation, at this stage, ignores gravity, and since the bullet otherwise satisfies the assumptions of the air to ground weapon, provided that the x' axis is parallel to the surface of the earth, equation (19) can be modified to give

$$A_{x'} = -E * V_{x'} \quad (21).$$

Since the velocity of the bullet during its time of flight will be several times larger than the speed of sound, and since the shape of the K_d vs. Mach curve is almost flat above 1.6 times the speed of sound, equation (21) can be further simplified to

$$A_{x'} = -p * K * V_{x'}^2 \quad (22)$$

or equivalently, since $V_{x'} > 0$,

$$A_{x'} / V_{x'}^2 = -p * K.$$

Since the effect of gravity is ignored, the density, p , will remain constant so that (22) becomes

$$A_{x'} / V_{x'}^2 = -K_2, \text{ a constant.} \quad (23)$$

Integrating both sides with respect to t gives

$$-1/V_{x'} = -K_2 * t + C_1. \quad (24)$$

Solving (24) for the initial conditions of the bullet as it leaves the gun gives

$1/V_{x'} = (K_2 * V_m * t + 1) / V_m$ (25) where V_m was taken to be $3300 + V_f$, and V_f was the fighter's velocity in feet per second. Expressing (25) in an equivalent form yields

$$V_{x'} = V_m / (K_2 * V_m * t + 1). \quad (26)$$

Integrating equation (26) again with respect to t gives the equation for distance traveled by the bullet

$$x' = \ln(K_2 * V_m * t + 1)/K_2 + C_3. \quad (27)$$

Since x' at $t=0$ is zero, $C_3=0$.

The time of flight may now be calculated as the time, t_{tof} , when

$$RANGE + (V_f - V_c) * t_{tof} = \ln(K_2 * V_m * t_{tof} + 1)/K_2. \quad (28)$$

RANGE is the distance to the target at time $t=0$. V_c is the closing velocity at time $t=0$.

Since the argument of the natural logarithm in equation (28) is strictly greater than 1, a Newton's fixed point iteration function may be used to find the value of t_{tof} . The iteration function is

$$t_{n+1} = t_n - F(t_n)/F'(t_n). \quad (29)$$

$F(t)$ and $F'(t)$ were defined as follows:

$$F(t) = RANGE + (V_f - V_c) * t - \ln(V_m * K_2 * t + 1)/K_2 \quad (30)$$

and

$$F'(t) = (V_f - V_c) - V_m / (V_m * K_2 * t + 1). \quad (31)$$

The first approximation of t_{tof} was used to find the values VX' , and subsequently ψ . To reduce the time

required to calculate t_2 , t_1 was always assigned the value one.

3. The Effect of Gravity Drop

In calculating the lead for target motion, the effect of gravity was ignored. A true trajectory, however, would always curve toward the earth as described by equation (20). For a time of flight less than two seconds, equation (20) may be simplified by assuming that $E=0$. This allows a closed form solution for the distance travelled by the bullet below the no gravity flight path during time, t_{tof} , seconds. The solution is

$$y = 32.17 * t^2 / 2 \quad (32)$$

The error in (32) is less than 20 feet reference [8].

The angle, in milliradians, subtended by the distance y_{tof} , at a future range of x'_{tof} feet is

$$\text{mils gravity drop} = y_{tof} * 1000 / x'_{tof} .$$

4. Effect of Angle of Attack

As the bullet leaves the barrel of the gun, it aligns with the relative wind. That is, the velocity vector of the bullet shifts from the gun boresight line to the x' axis. The amount of this shift, in mils, is calculated as

$$\text{MILS}_{\text{AOA}} = V_f * \text{ACA} / (V_f + 3300),$$

where AOA represents angle of attack, in mils, of the gun barrel boresight line.

5. Parallax Correction

Since the gun and the sight pivot are approximately seven feet apart in the z' direction, a small factor must be added to the sight setting. In the F-4, this setting is 3.19 mils, reference [8]. Fig 9 shows that the effect of adding a parallax correction is to have the zero sight line (ZSL) intersect the gun boresight line (TBL) at a known distance (harmonization range) in front of the fighter.

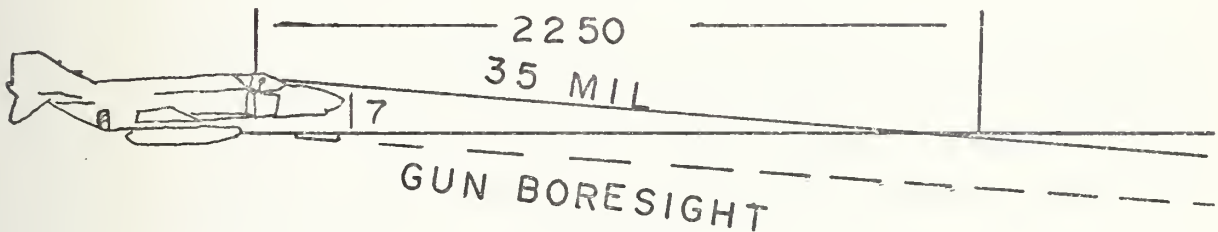


Fig 9 PARALLAX CORRECTION

6. Implementation and Timing

The calculations described above were coded into a PLM subroutine and tested on the INTERP8 simulation subroutine. The time of execution was found to be less than .2 seconds. In addition, a simulated sampling and data preparation routine was timed at approximately .15 seconds. See appendix H.

V. CONCLUSIONS

Based on the experiments performed and the timing data obtained, the micro computer is functionally capable of providing an improvement in air to ground delivery accuracy, which is on the order of a two fold increase in accuracy. Since the cost of micro computers is decreasing, whereas the cost of fuel and replacement aircraft is increasing, and since improved delivery accuracy means fewer sorties and aircraft exposed to hostile fire, a program to study the engineering factors and actual airborne effectiveness of the proposed micro computer modification is warranted.

The air to air gunnery implementation was not compared to the fixed sight solution since there is no criteria by which to judge the accuracy of the pilot's guess in the latter case. The reaction time of the computer (less than .5 second) appears to be adequate.

Although no live tests were conducted on the system in either mode, such tests could be conducted at a relatively low cost by programming an A-6E or A-7E airborne computer to simulate the accuracy and timing of the proposed micro computer system. This would eliminate the need to reconfigure the hardware of the test aircraft, at least in the preliminary stages.

The cost of the micro computer and associated 10,006 bytes of memory necessary to execute the air to ground and air to air programs is one thousand one hundred and fifty dollars based on currently available prices. Overall, the

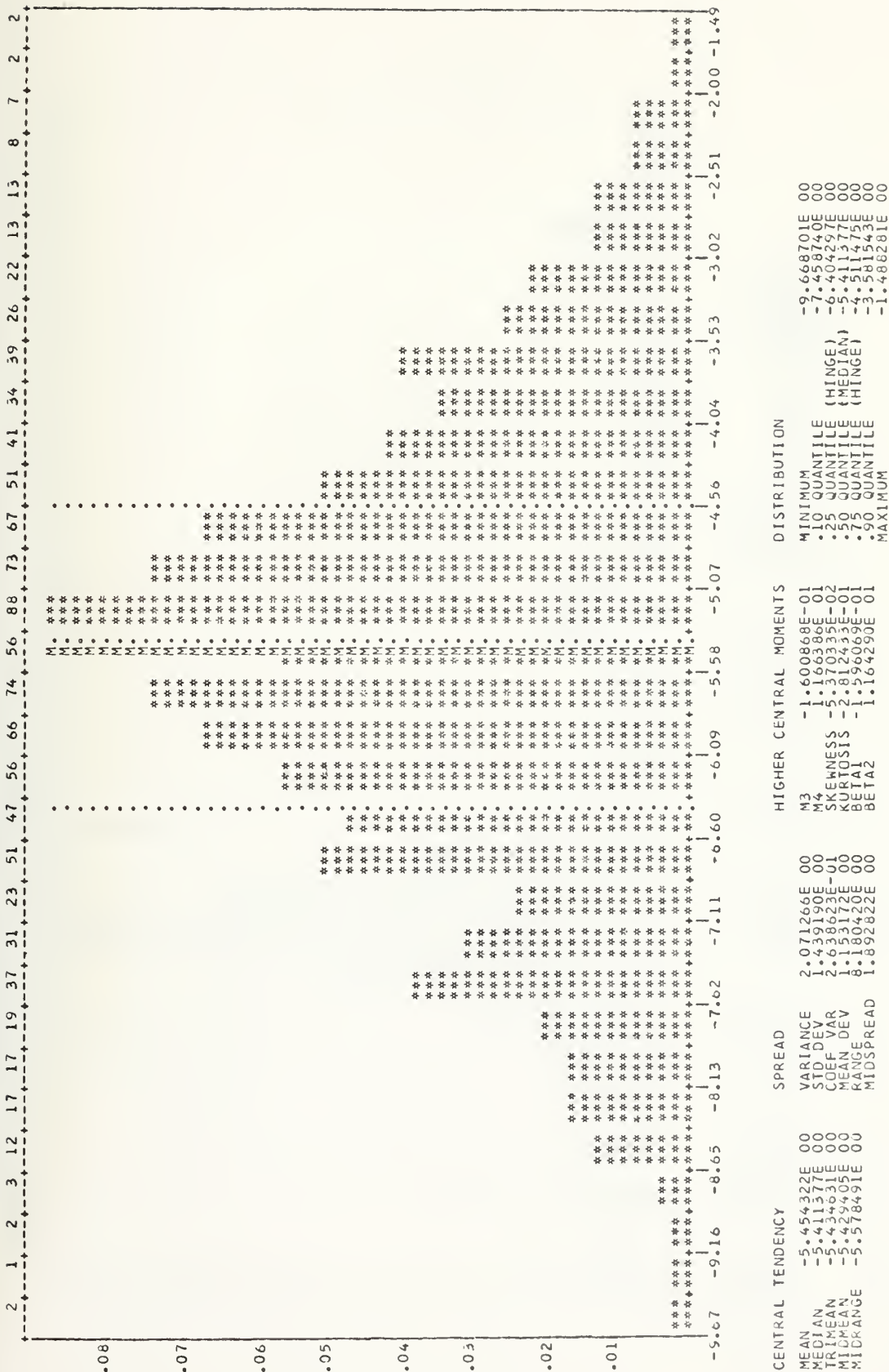
micro computer represents one way to help curb the rapid increase in the cost of aircraft procurement.

APPENDIX A

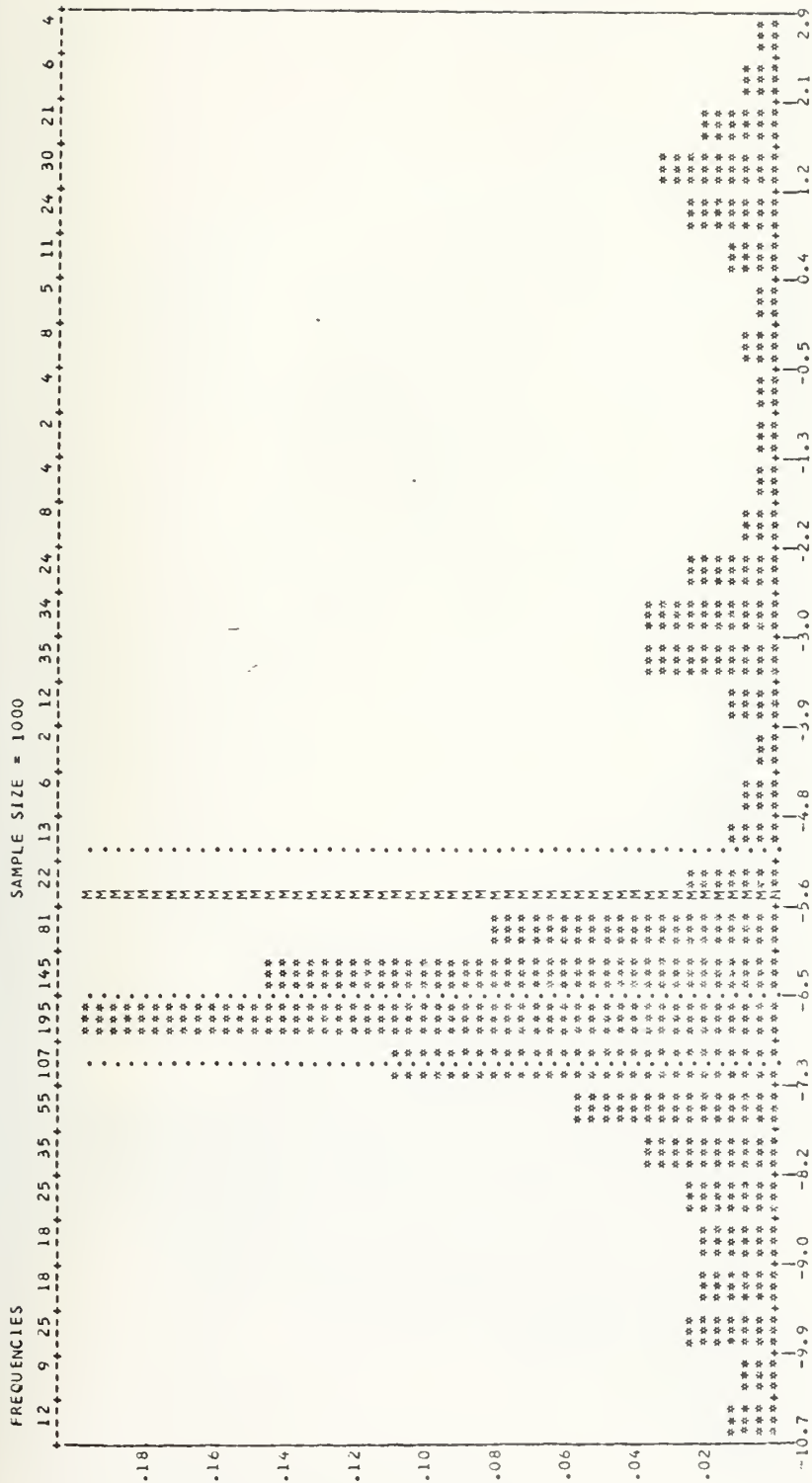


CENTRAL TENDENCY		SPREAD		HIGHER CENTRAL MOMENTS		DISTRIBUTION	
MEAN	-5.771431E 00	VARIANCE	3.583900E 00	M3	-1.332822E 00	MINIMUM	-1.105664E 01
MEDEVIAN	-5.786273E 00	STD DEV	1.893119E 00	M4	-1.387182E 01	QUANTILE	-8.478032E 00
MEAN	-5.780929E 00	COEF	3.209475E 01	SKENESS	-2.963282E 01	QUANTILE	-5.984739E 00
MEDEVIAN	-5.784050E 00	RANGE	1.052217E 01	BETA1	-3.371922E 01	QUANTILE	-3.686398E 00
MDRANGE	-3.784050E 00	MDSREAD	2.451358E 00	BETA2	3.371922E 01	QUANTILE	-3.332391E 00
						MAXIMUM	-5.114746E 01

DISTRIBUTION OF ERRORS IN PREDICTED AIRSPEED USING ORTHOGONAL POLYNOMIAL



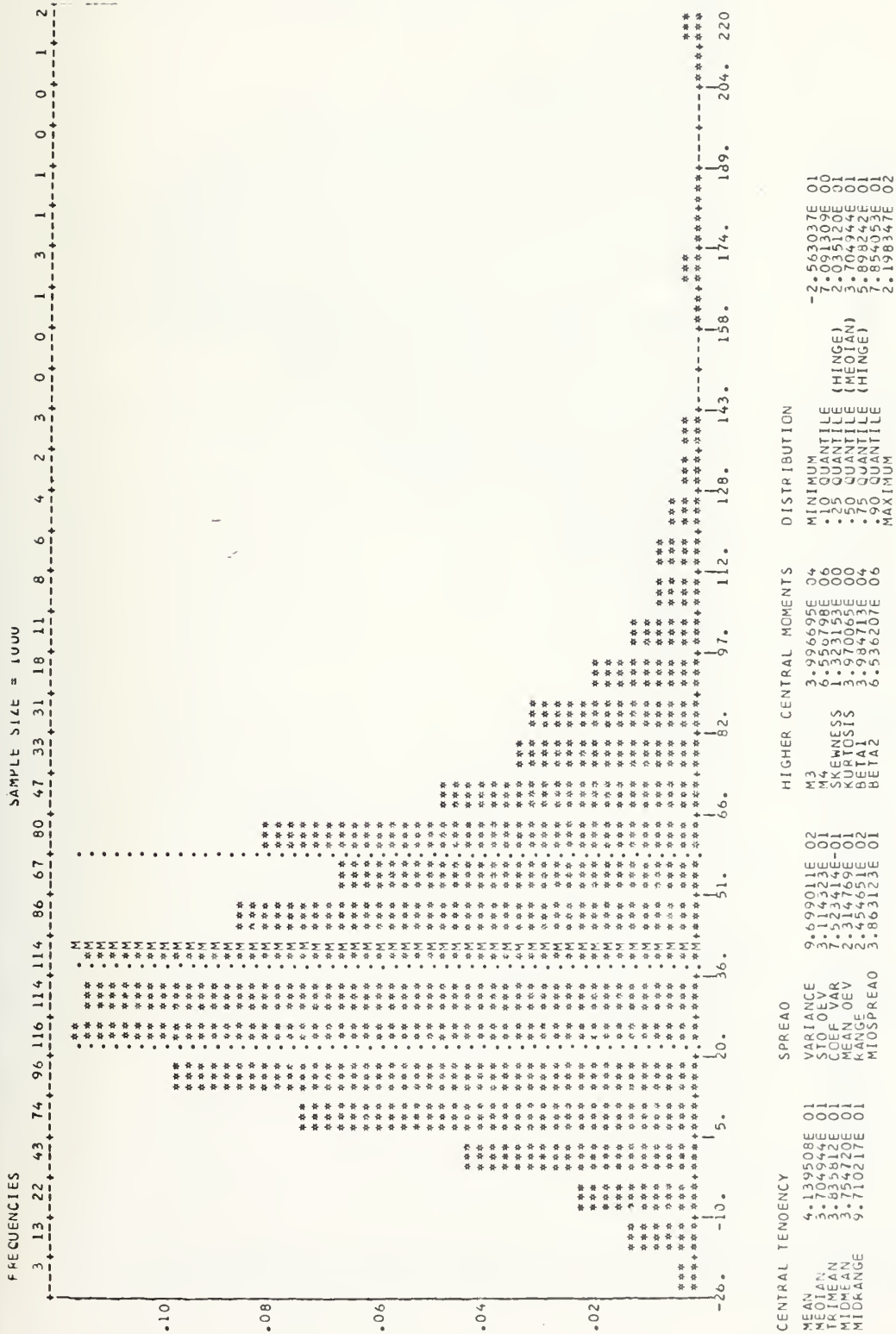
ERRORS IN PREDICTED AIRSPEED USING AVERAGED INPUT

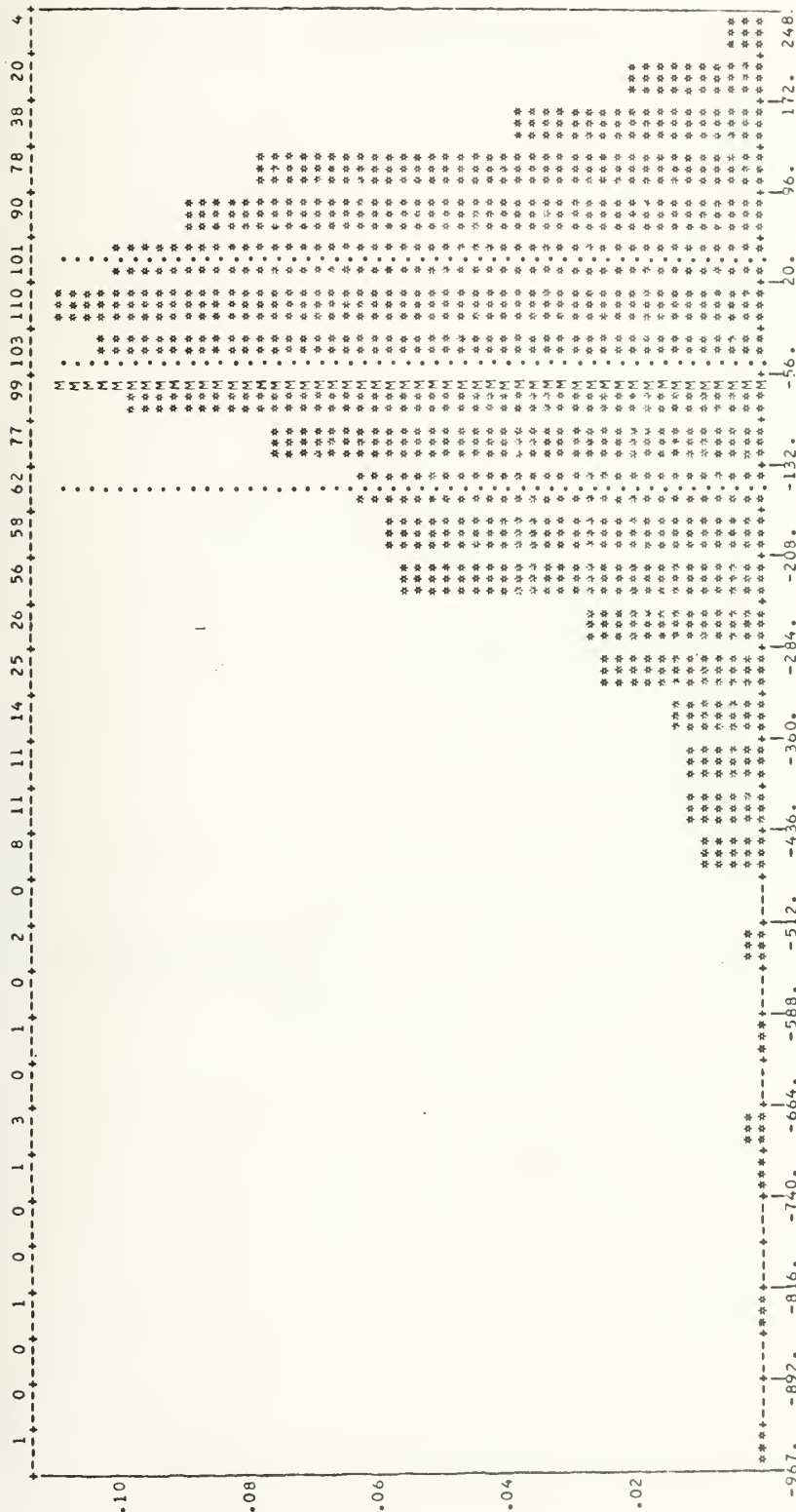


CENTRAL TENDENCY		SPREAD		HIGHER CENTRAL MOMENTS		DISTRIBUTION	
MEAN	-5.55659E 00	VARIANCE	8.70713E 00	M3	3.10644E 01	MINIMUM	-1.07463E 01
STDEV	-6.75587E 00	STD DEV	2.95037E 00	M4	2.89628E 02	.10 QUANTILE	-1.81183E 00
MEAN	-6.59827E 00	MEAN DEV	1.56409E 00	SKENESS	1.50092E -01	.25 QUANTILE	-1.5923E 00
MIDRANGE	-3.60133E 00	RANGE	1.80891E 01	BETA1	3.40913E 01	.50 QUANTILE	-1.5253E -03
		MIDSPREAD	2.01831E 00	BETA2	2.88424E 02	.75 QUANTILE	5.61523E 00
						.90 QUANTILE	2.94287E 00
						MAXIMUM	

ERRORS IN PREDICTED AIRSPEED USING ONE SECOND SAMPLES

APPENDIX B





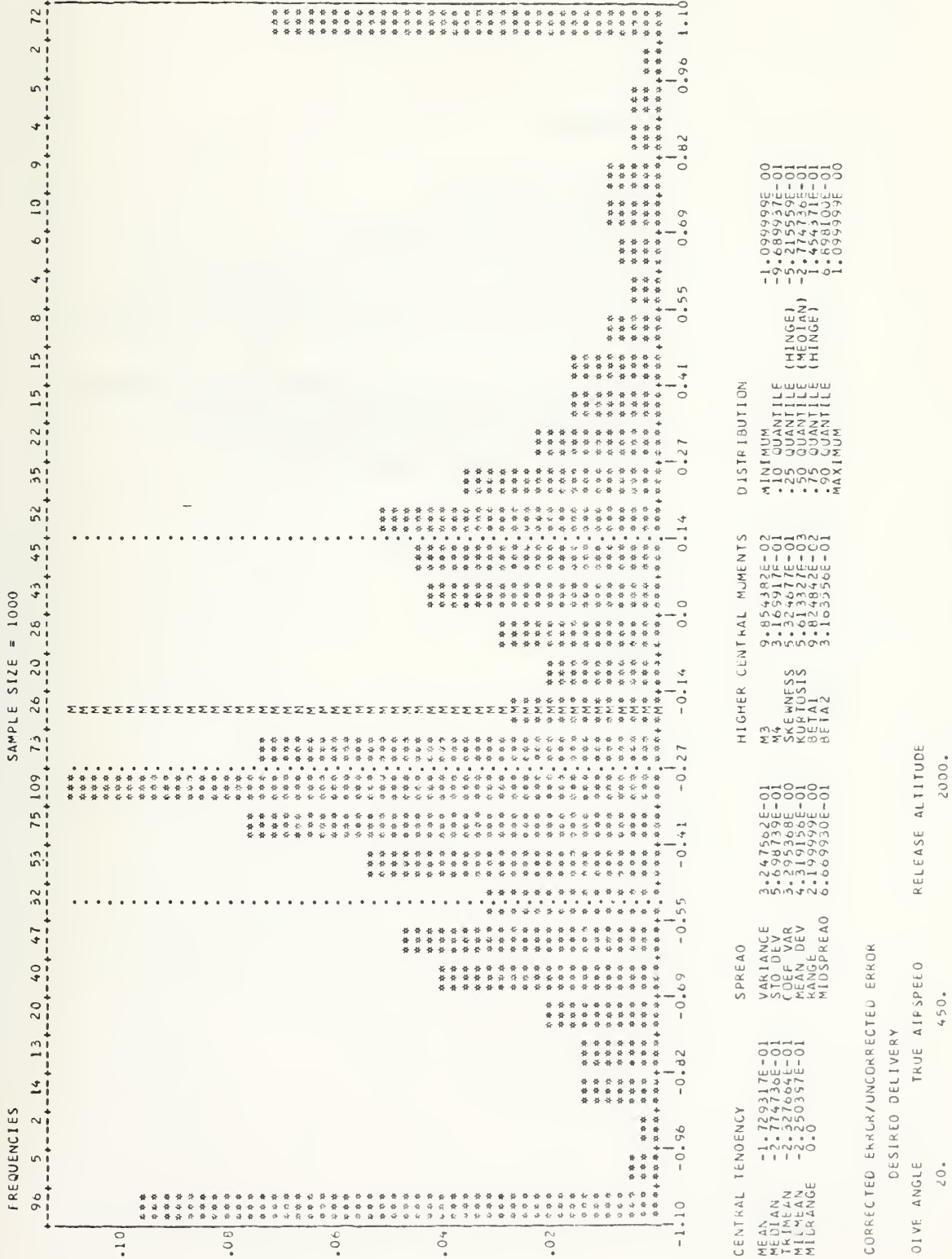
CENTRAL TENDENCY		SPREAD		HIGHER CENTRAL MOMENTS		DISTRIBUTION	
MEAN	-6.071706E 01	VARIANCE	2.326854E 04	M3	-3.694105E 09	MINIMUM	-9.674492E 02
MEDIAN	-4.264950E 01	STD DEV	1.525403E 02	M4	2.918793E 09	10 QUANTILE	-2.490649E 02
TRIMEAN	-4.656146E 01	COEF VAR	2.512313E 00	SKEWNESS	-1.040771E 00	.25 QUANTILE	-1.427869E 02
MIDMEAN	-4.244736E 01	MEAN DEV	1.172591E 02	KURTOSIS	2.390944E 00	.50 QUANTILE (MEDIAN)	-7.864908E 01
MIDRANGE	-3.599658E 02	RANGE	1.244967E 03	BETA1	-2.683031E 00	.75 QUANTILE	7.854032E 01
		MIDSPREAD	1.986277E 02	BETA2	2.910361E 09	.90 QUANTILE	2.475173E 02
						MAXIMUM	

ERROR IN UNCORRECTED OCMN RANGE TRAVEL

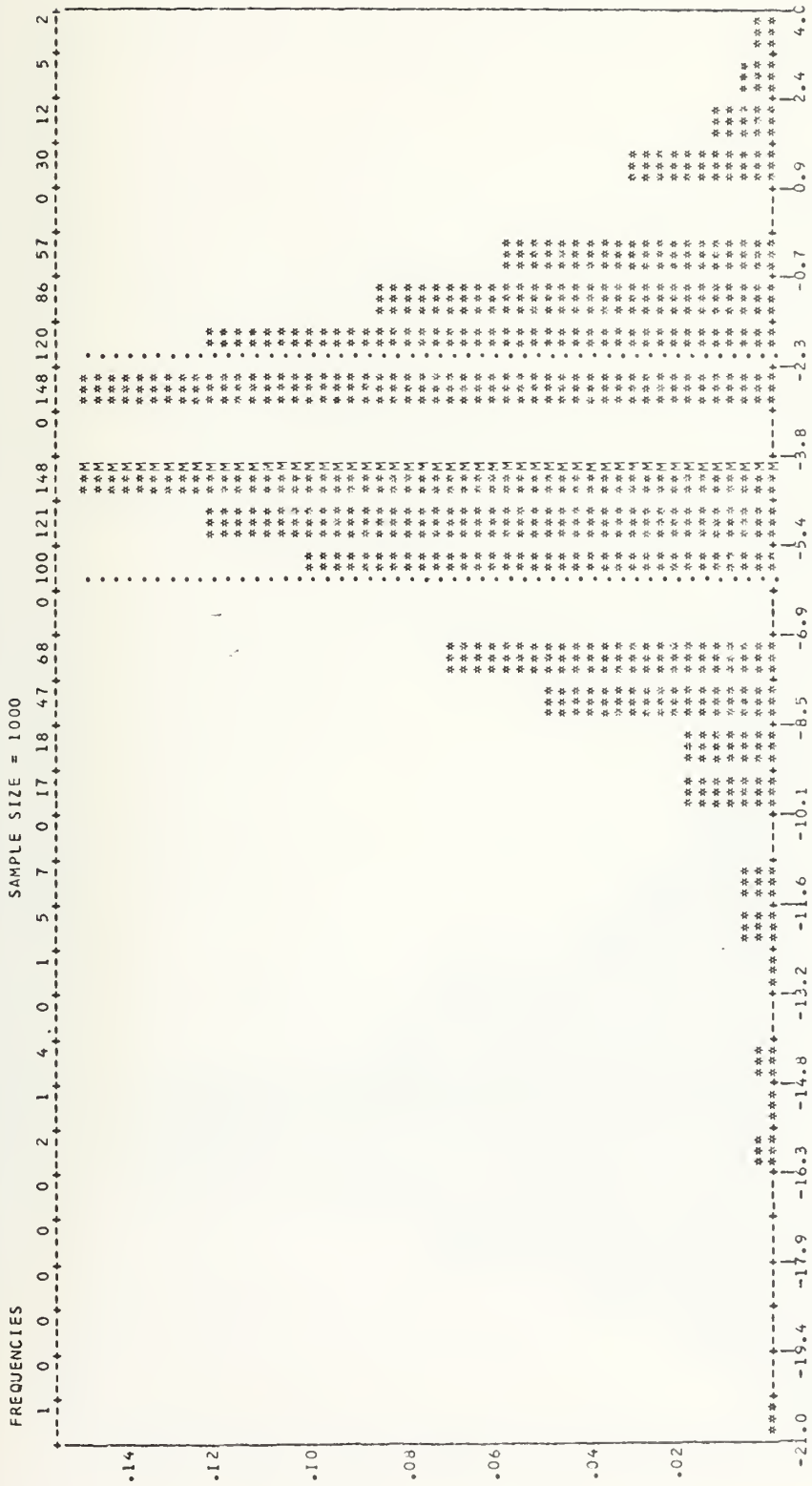
DESIREO DELIVERY

DIVE ANGLE 20. RELEASE ALTITUDE 2000.

TRUE AIRSPEED 450.



SAMPLE SIZE = 1000



CENTRAL TENDENCY		SPREAD		HIGHER CENTRAL MOMENTS		DISTRIBUTION	
MEAN	-3.948000E 00	VARIANCE	8.663958E 00	M3	-1.631726E 01	MINIMUM	-2.100000E 01
MEDIAN	-4.000000E 00	STD DEV	2.943460E 00	M4	3.583765E 02	.10 QUANTILE	-8.000000E 00
TRIMEAN	-4.000000E 00	COEF VAR	7.455578E -01	SKENESS	-6.398420E -01	.25 QUANTILE	-6.000000E 00
MIDMEAN	-3.782000E 00	MEAN DEV	2.242000E 00	KURTOSIS	-1.774269E 01	.50 QUANTILE	-3.000000E 00
MIDRANGE	-8.500000E 00	RANGE	2.500000E 01	BETA1	-1.626834E 01	.75 QUANTILE	-2.000000E 00
		MIDSPREAD	4.000000E 00	BETA2	3.573928E 02	MAXIMUM	4.000000E 00

ERROR IN SIGHT SETTING
DESIRED DELIVERY
DIVE ANGLE 26.
TRUE AIRSPEED 450.
RELEASE ALTITUDE 2000.

SAMPLE SIZE = 1000

FREQUENCIES

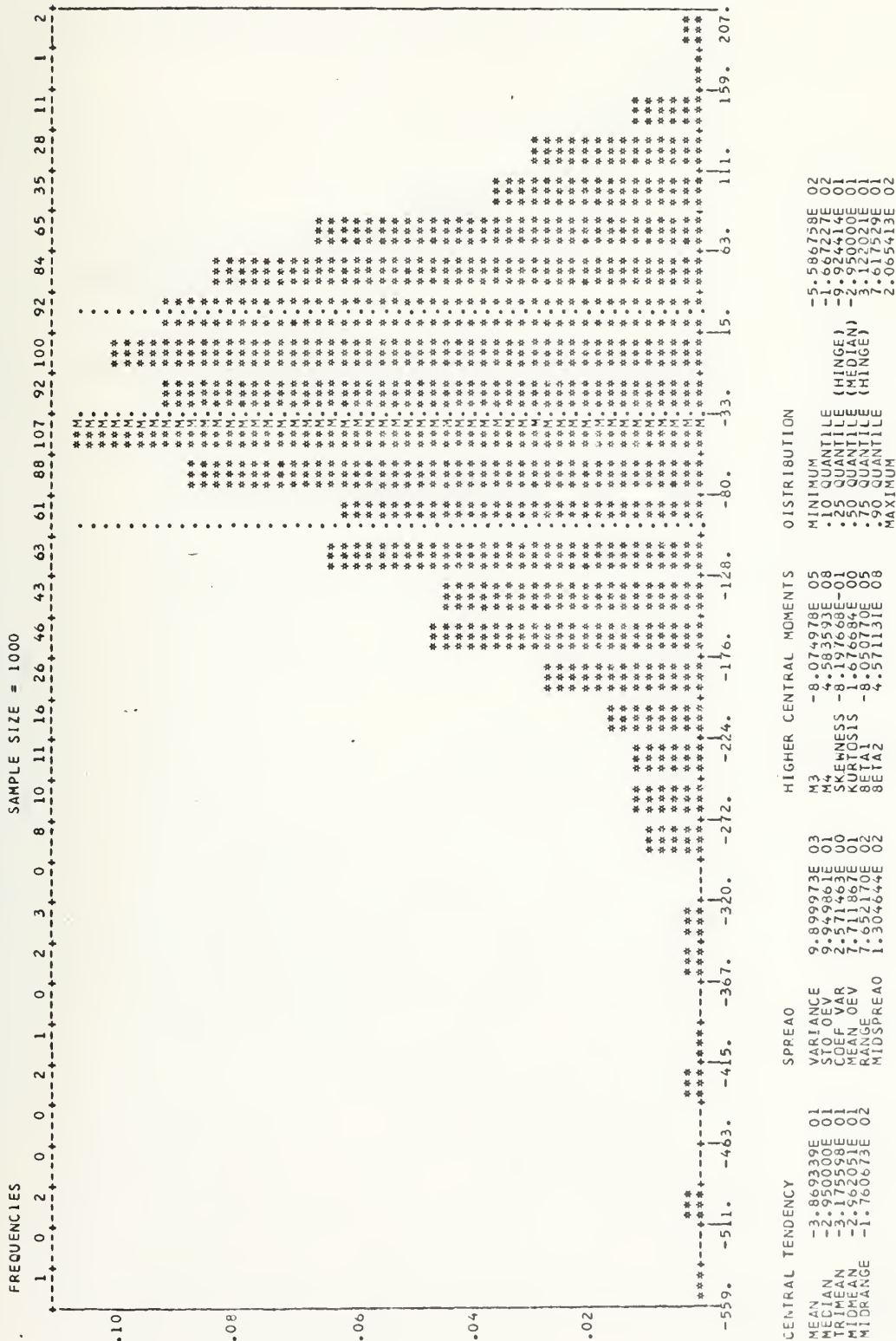


CENTRAL TENDENCY		SPREAD		HIGHER CENTRAL MOMENTS		DISTRIBUTION	
MEAN	-2.90000E 00	VARIANCE	3.78764E 02	M2	7.81766E 03	MINIMUM	-1.94502E 01
MEDIAN	-5.32998E 00	STD DEV	1.94619E 01	M3	9.02639E 03	QUANTILE	-1.25501E 01
TRIMEAN	-4.55089E 00	CORR VAR	6.71098E 00	SKENESS	1.02639E 00	QUANTILE	-1.25501E 01
MICMEAN	-4.66219E 00	MEAN DEV	1.48108E 01	KURTOSIS	3.890279E 00	QUANTILE	-1.25501E 01
MIDRANGE	4.57561E 01	RANGE	1.90412E 02	BET1	7.798230E 03	QUANTILE	2.12890E 01
		MIDSPREAD	2.37475E 01	BET2	9.798661E 05	QUANTILE	1.40962E 02
						MAXIMUM	

ERFOR IN CCWN RANGE TRAVEL

DESIRED DELIVERY

DIVE ANGLE 30. TRUE AIRSPEED 450. RELEASE ALTITUDE 3000.

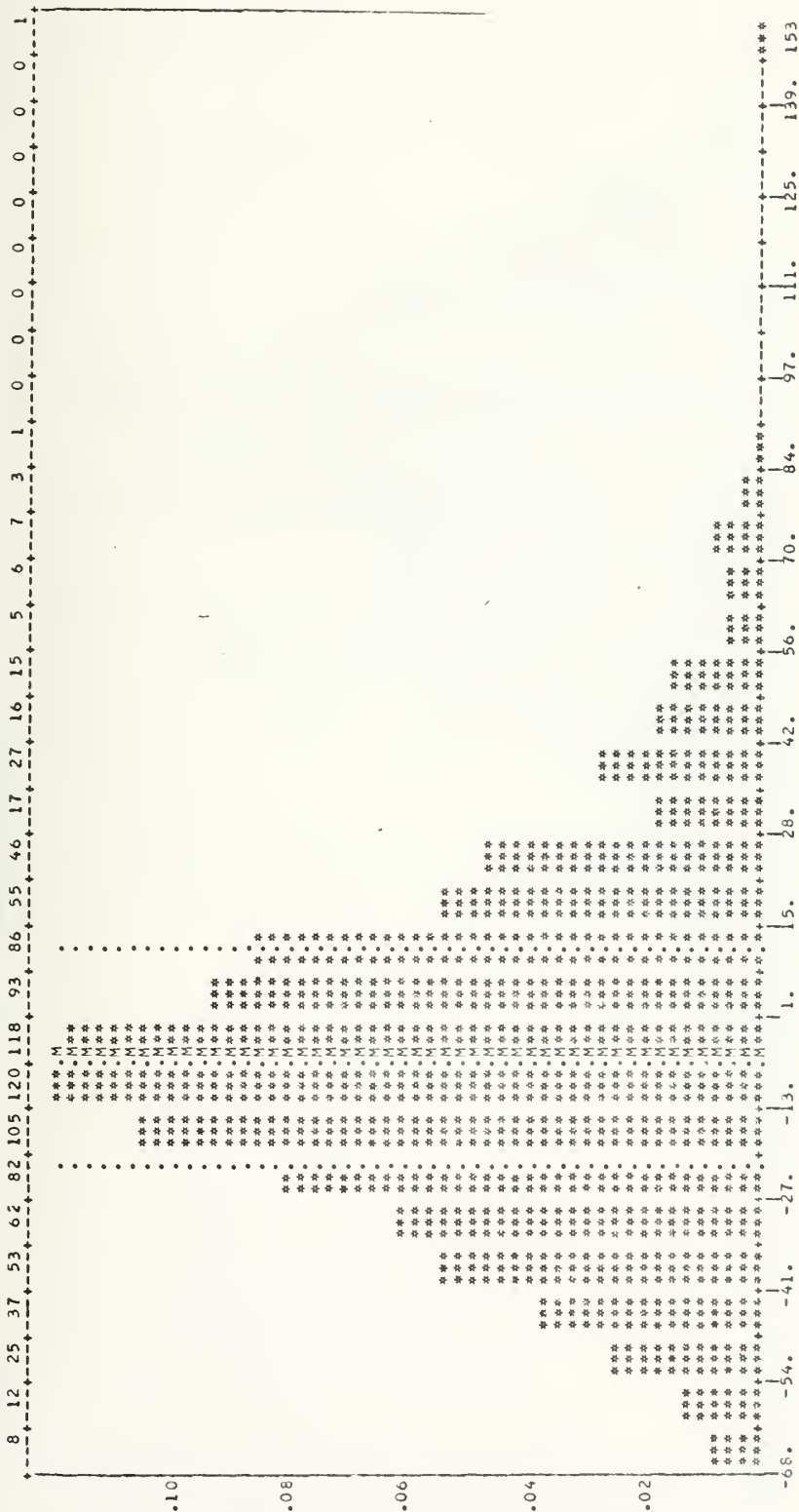


ERROR IN UNCORRECTED OCWN RANGE TRAVEL

DESIRED DELIVERY

OIVE ANGLE TRUE AIRSPEED RELEASE ALTITUDE

30. 450. 3000.



CENTRAL TENDENCY	SPREAD	HIGHER CENTRAL MOMENTS	DISTRIBUTION
MEAN	00	M3	M10
MODE	00	M4	M9
MEAN	00	M5	M8
MODE	00	M6	M7
MEAN	00	M7	M6
MODE	00	M8	M5
MEAN	00	M9	M4
MODE	00	M10	M3
MEAN	00	M11	M2
MODE	00	M12	M1
MEAN	00	M13	M0
MODE	00	M14	M0
MEAN	00	M15	M0
MODE	00	M16	M0
MEAN	00	M17	M0
MODE	00	M18	M0
MEAN	00	M19	M0
MODE	00	M20	M0
MEAN	00	M21	M0
MODE	00	M22	M0
MEAN	00	M23	M0
MODE	00	M24	M0
MEAN	00	M25	M0
MODE	00	M26	M0
MEAN	00	M27	M0
MODE	00	M28	M0
MEAN	00	M29	M0
MODE	00	M30	M0
MEAN	00	M31	M0
MODE	00	M32	M0
MEAN	00	M33	M0
MODE	00	M34	M0
MEAN	00	M35	M0
MODE	00	M36	M0
MEAN	00	M37	M0
MODE	00	M38	M0
MEAN	00	M39	M0
MODE	00	M40	M0
MEAN	00	M41	M0
MODE	00	M42	M0
MEAN	00	M43	M0
MODE	00	M44	M0
MEAN	00	M45	M0
MODE	00	M46	M0
MEAN	00	M47	M0
MODE	00	M48	M0
MEAN	00	M49	M0
MODE	00	M50	M0
MEAN	00	M51	M0
MODE	00	M52	M0
MEAN	00	M53	M0
MODE	00	M54	M0
MEAN	00	M55	M0
MODE	00	M56	M0
MEAN	00	M57	M0
MODE	00	M58	M0
MEAN	00	M59	M0
MODE	00	M60	M0
MEAN	00	M61	M0
MODE	00	M62	M0
MEAN	00	M63	M0
MODE	00	M64	M0
MEAN	00	M65	M0
MODE	00	M66	M0
MEAN	00	M67	M0
MODE	00	M68	M0
MEAN	00	M69	M0
MODE	00	M70	M0
MEAN	00	M71	M0
MODE	00	M72	M0
MEAN	00	M73	M0
MODE	00	M74	M0
MEAN	00	M75	M0
MODE	00	M76	M0
MEAN	00	M77	M0
MODE	00	M78	M0
MEAN	00	M79	M0
MODE	00	M80	M0
MEAN	00	M81	M0
MODE	00	M82	M0
MEAN	00	M83	M0
MODE	00	M84	M0
MEAN	00	M85	M0
MODE	00	M86	M0
MEAN	00	M87	M0
MODE	00	M88	M0
MEAN	00	M89	M0
MODE	00	M90	M0
MEAN	00	M91	M0
MODE	00	M92	M0
MEAN	00	M93	M0
MODE	00	M94	M0
MEAN	00	M95	M0
MODE	00	M96	M0
MEAN	00	M97	M0
MODE	00	M98	M0
MEAN	00	M99	M0
MODE	00	M100	M0

ERROR IN DCWN RANGE TRAVEL

DESIRED DELIVERY

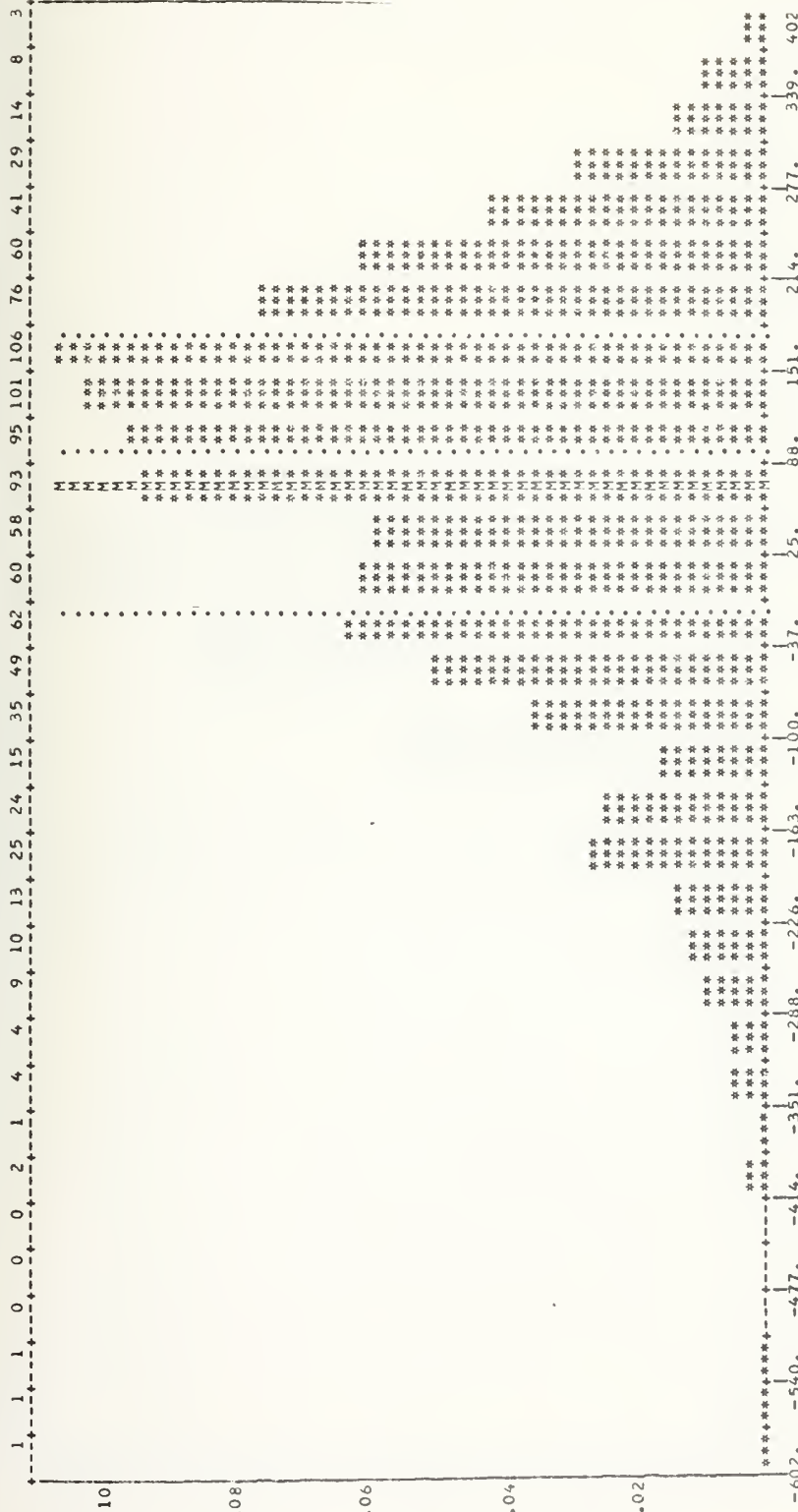
DIVE ANGLE TRUE AIRSPEED RELEASE ALTITUDE

30. 500.

5000.

SAMPLE SIZE = 1000

FREQUENCIES

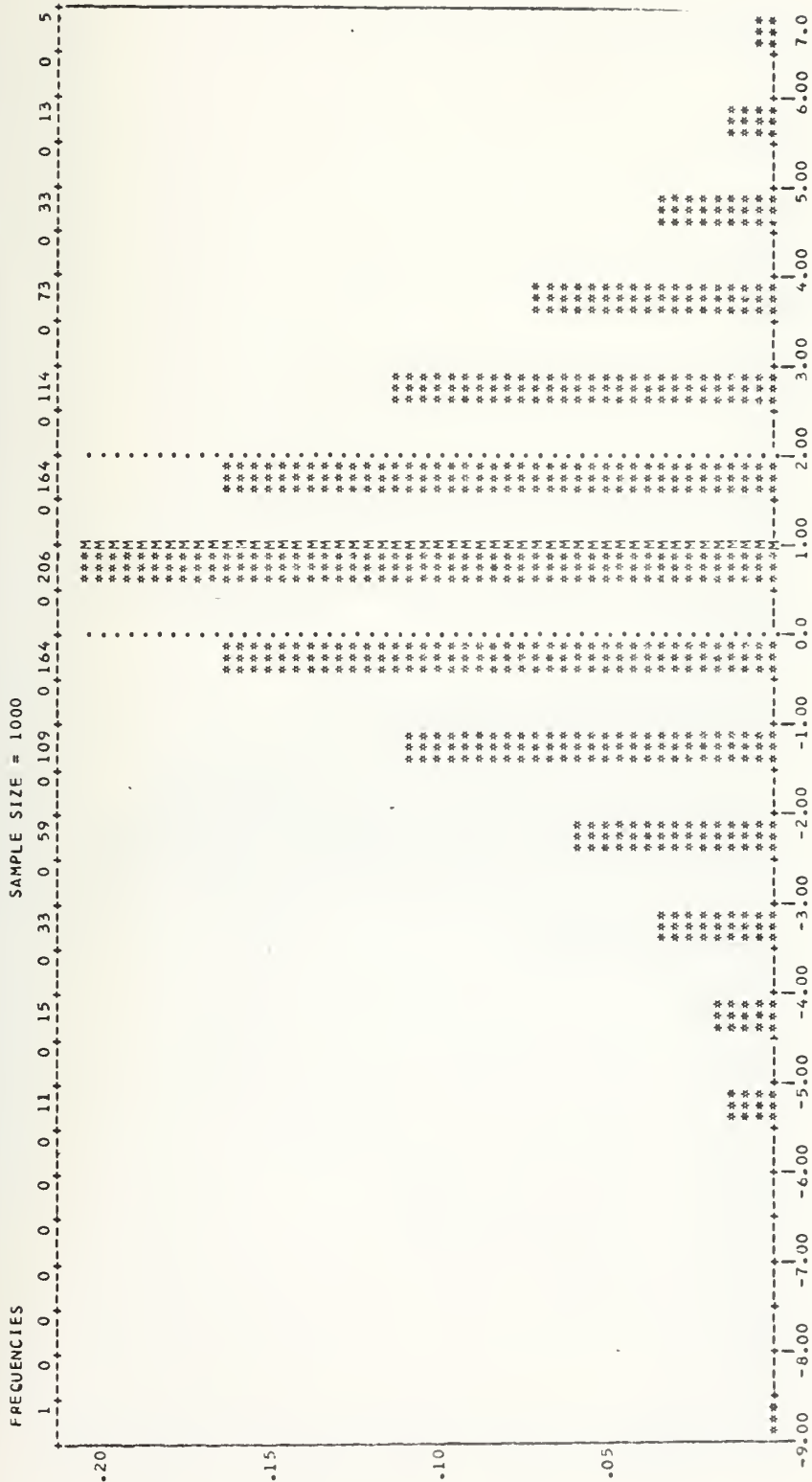


CENTRAL TENDENCY		SPREAD		HIGHER CENTRAL MOMENTS		DISTRIBUTION	
MEAN	7.742320E 01	VARIANCE	2.073498E 04	M3	-2.290636E 06	MINIMUM	-6.022813E 02
STDEV	9.868164E 01	STD DEV	1.439965E 02	M4	-1.689853E 09	.10 QUANTILE	-1.166680E 02
COEF VAR	9.332910E 01	COEF VAR	1.859801E 00	SKEWNESS	-7.671861E-01	.25 QUANTILE	-9.408203E 00
MEAN DEV	9.442783E 01	MEAN DEV	1.113410E 02	KURTOSIS	9.304457E-01	.50 QUANTILE (MEDIAN)	1.773913E 02
RANGE	-1.001074E 02	RANGE	1.004348E 03	BETA1	-2.283769E 09	.75 QUANTILE	2.433398E 02
MIDRANGE	-1.001074E 02	MIDRANGE	1.867695E 02	BETA2	1.685666E 09	.90 QUANTILE	4.020664E 02
						MAXIMUM	

ERROR IN UNCORRECTED DOWN RANGE TRAVEL

DESIRED DELIVERY

DIVE ANGLE TRUE AIRSPEED RELEASE ALTITUDE 5000.

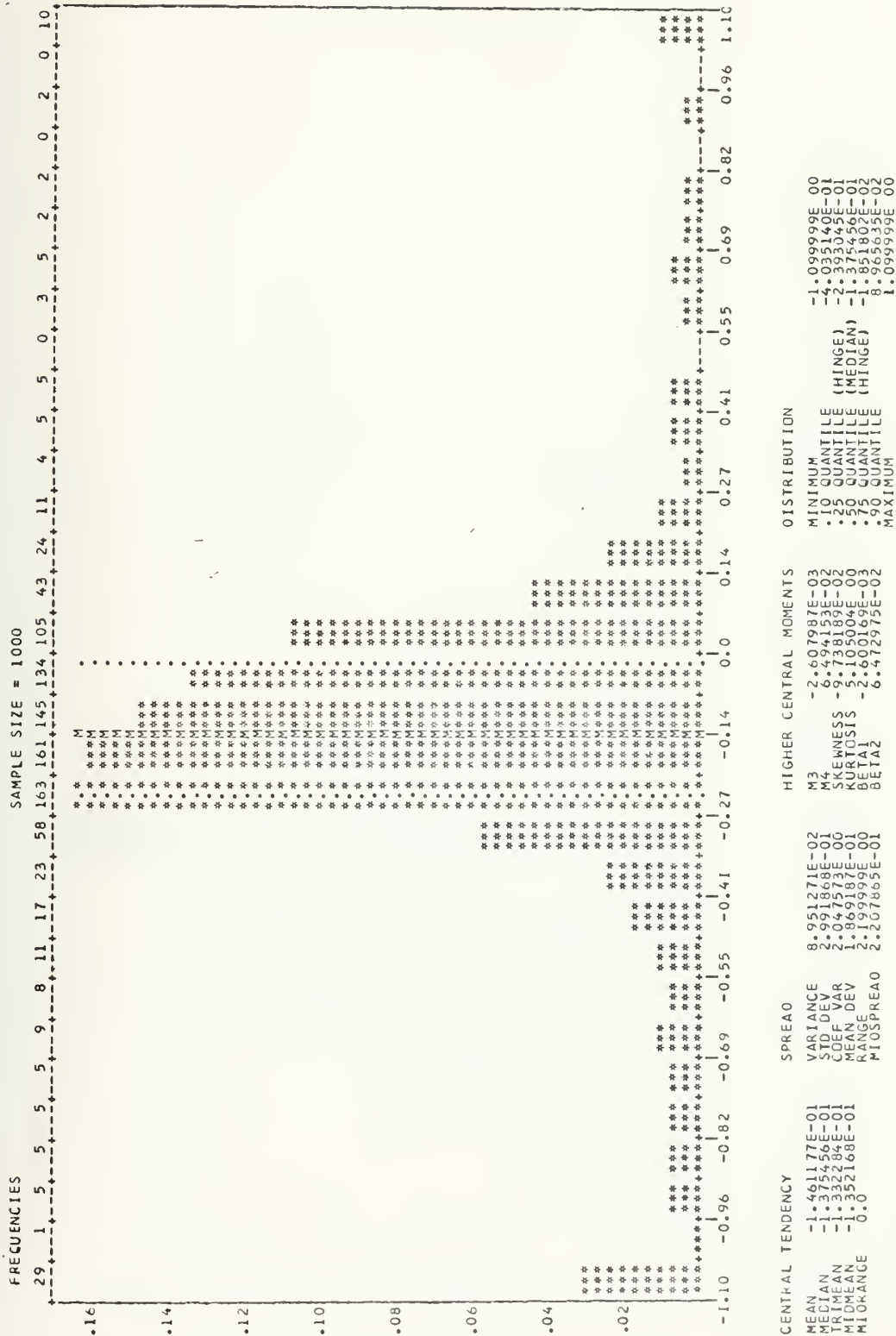


CENTRAL TENDENCY		SPREAD		HIGHER CENTRAL MOMENTS		DISTRIBUTION	
MEAN	9.9600000E-01	VARIANCE	4.844827E 00	M3	-2.076140E 00	MINIMUM	-9.000000E 00
MEDIAN	1.000000E 00	STD DEV	2.201097E 00	M4	7.986519E 01	.10 QUANTILE	-2.000000E 00
TRIMEAN	1.000000E 00	COEF VAR	2.209936E 00	SKEWNESS	-1.946881E-01	.25 QUANTILE	0.0
MIDMEAN	1.020000E 00	MEAN DEV	1.679999E 00	KURTOSIS	4.025249E-01	.50 QUANTILE (MEDIAN)	1.000000E 00
MIDRANGE	-1.000000E 00	RANGE	1.600000E 01	BETA1	-2.069917E 00	.75 QUANTILE	2.000000E 00
		MIDSPREAD	2.000000E 00	BETA2	7.9686634E 01	.90 QUANTILE	4.000000E 00
						MAXIMUM	7.000000E 00

ERROR IN SIGHT SETTING

DESIRED DELIVERY

DIVE ANGLE 30. TRUE AIRSPEED 500. RELEASE ALTITUDE 5000.





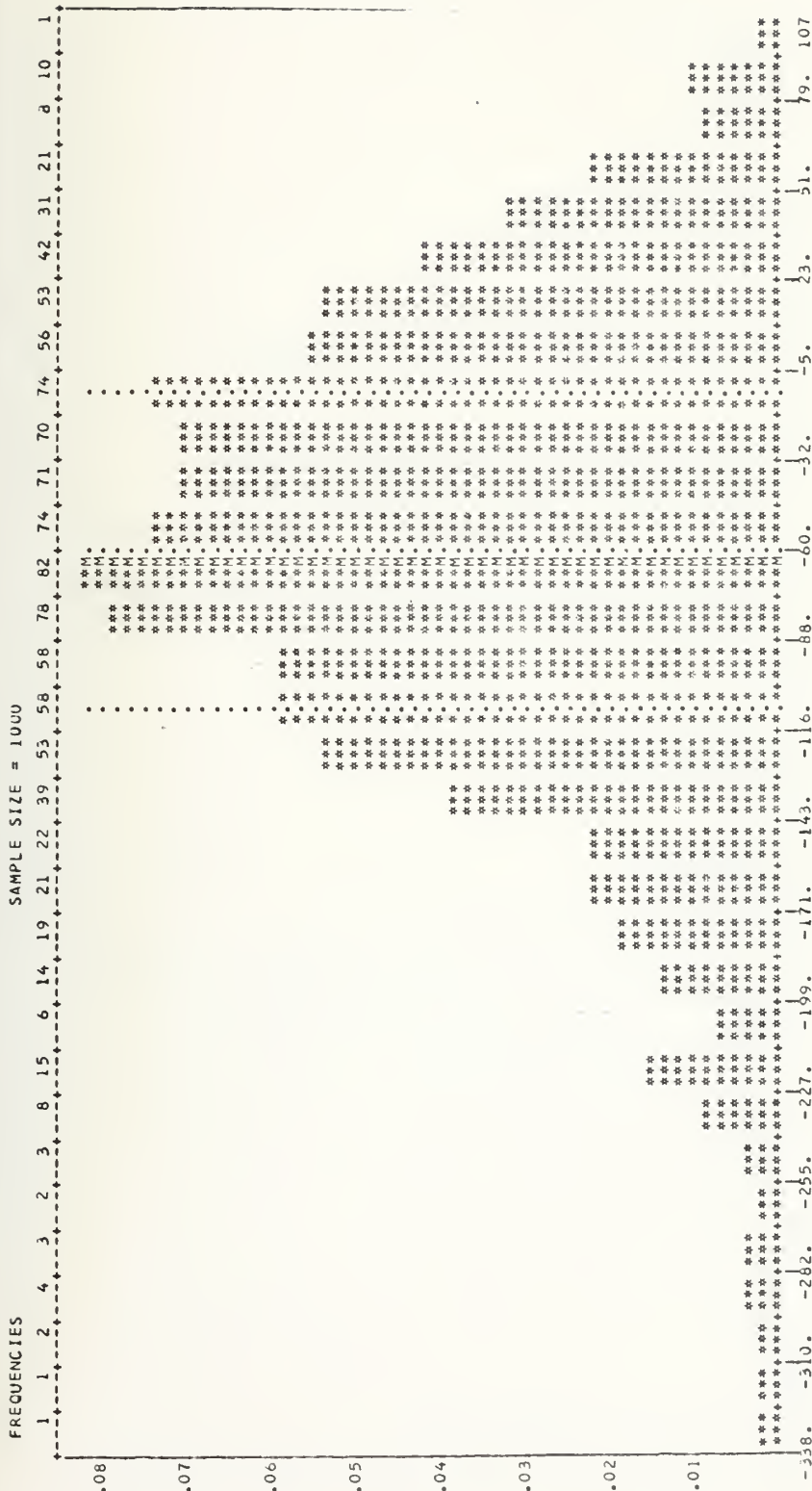
CENTRAL TENDENCY		SPREAD		HIGHER CENTRAL MOMENTS		DISTRIBUTION	
MEAN	3.162369E 01	VARIANCE	4.917185E 02	M3	6.316648E 03	MINIMUM	-1.623828E 01
STDEV	3.016345E 01	STD DEV	2.217473E 01	M4	8.643069E 05	.10 QUANTILE	4.468506E 00
MEAN	3.008444E 01	MEAN DEV	1.713509E 01	SKENESS	5.793111E -01	.25 QUANTILE	1.599719E 01
MIDRANGE	3.380212E 01	RANGE	1.405808E 02	BET11	8.299132E 03	.50 QUANTILE (MEDIAN)	3.016345E 01
		MIDSPREAD	2.851096E 01	BET12	8.622281E 05	.75 QUANTILE	5.939292E 01
						.90 QUANTILE	1.238425E 02
						MAXIMUM	

ERROR IN DCMN RANGE TRAVEL

DESIRED DELIVERY

DIVE ANGLE TRUE AIRSPEED RELEASE ALTITUDE

45. 450. 4000.



CENTRAL TENDENCY

MEAN -6.220845E 01
MEDIAN -5.843384E 01
TRIMEAN -5.885822E 01
MIDMEAN -5.718930E 01
MIDRANGE -1.156841E 02

SPREAD

VARIANCE 5.289051E 03
STD DEV 7.272586E 01
COEF VAR 1.169066E 00
MEAN DEV 5.734593E 01
RANGE 4.446260E 02
MIDSPREAD 9.576709E 01

HIGHER CENTRAL MOMENTS

M3 -2.122805E 05
M4 -9.505205E 07
SKENNESS -5.518785E-01
KURTOSIS 3.978653E-01
BETA1 -3.116441E 05
BETA2 9.489942E 07

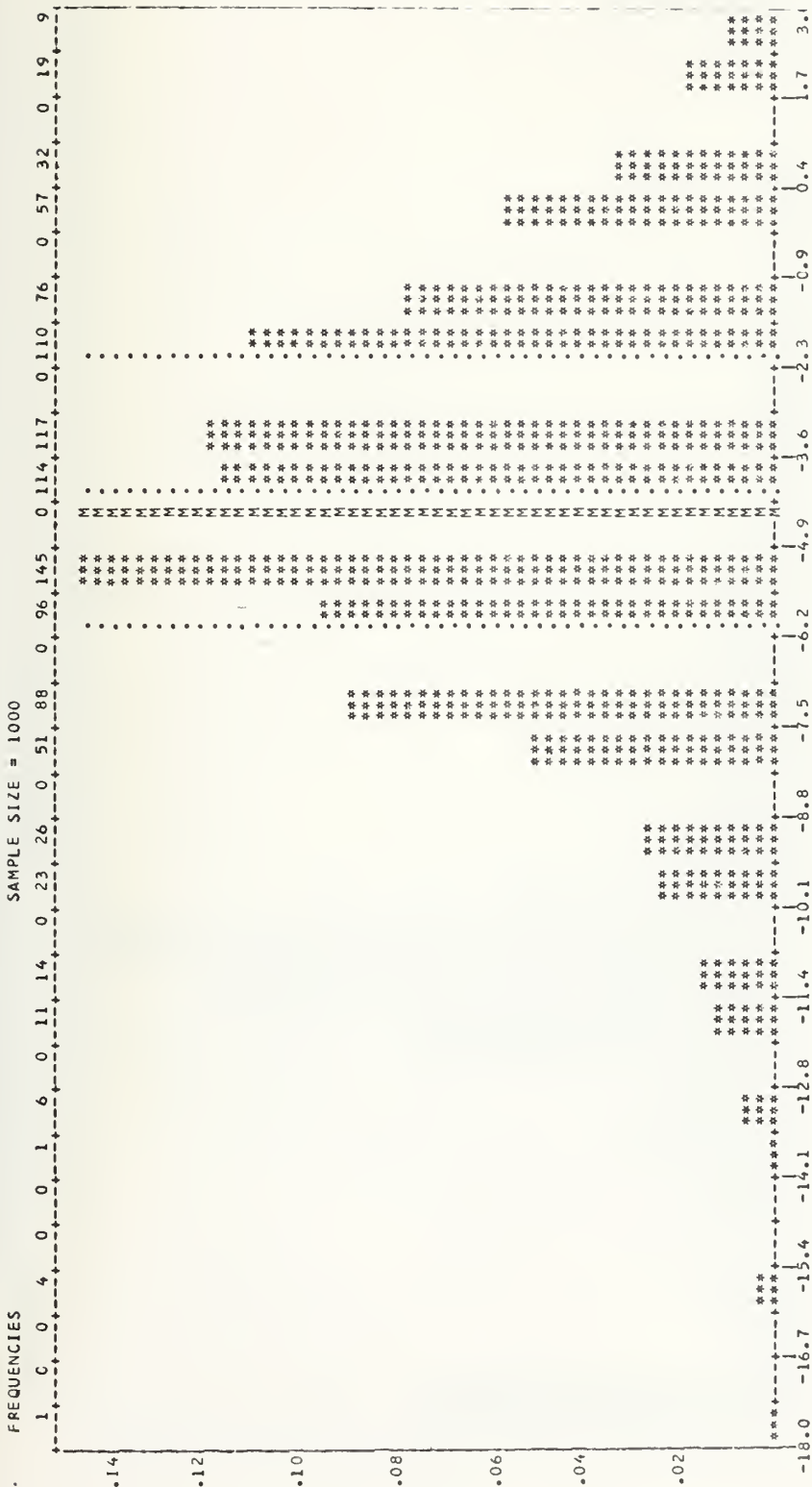
DISTRIBUTION

MINIMUM -3.379971E 02
10 QUANTILE -1.568501E 02
25 QUANTILE -1.071661E 02
50 QUANTILE (MEDIAN) -5.843384E 01
75 QUANTILE -1.139903E 01
90 QUANTILE -2.894238E 01
MAXIMUM 1.066289E 02

ERROR IN UNCORRECTED DOWN RANGE TRAVEL

DESIRE DELIVERY

DIVE ANGLE 45.
TRUE AIRSPEED 450.
RELEASE ALTITUDE 4000.

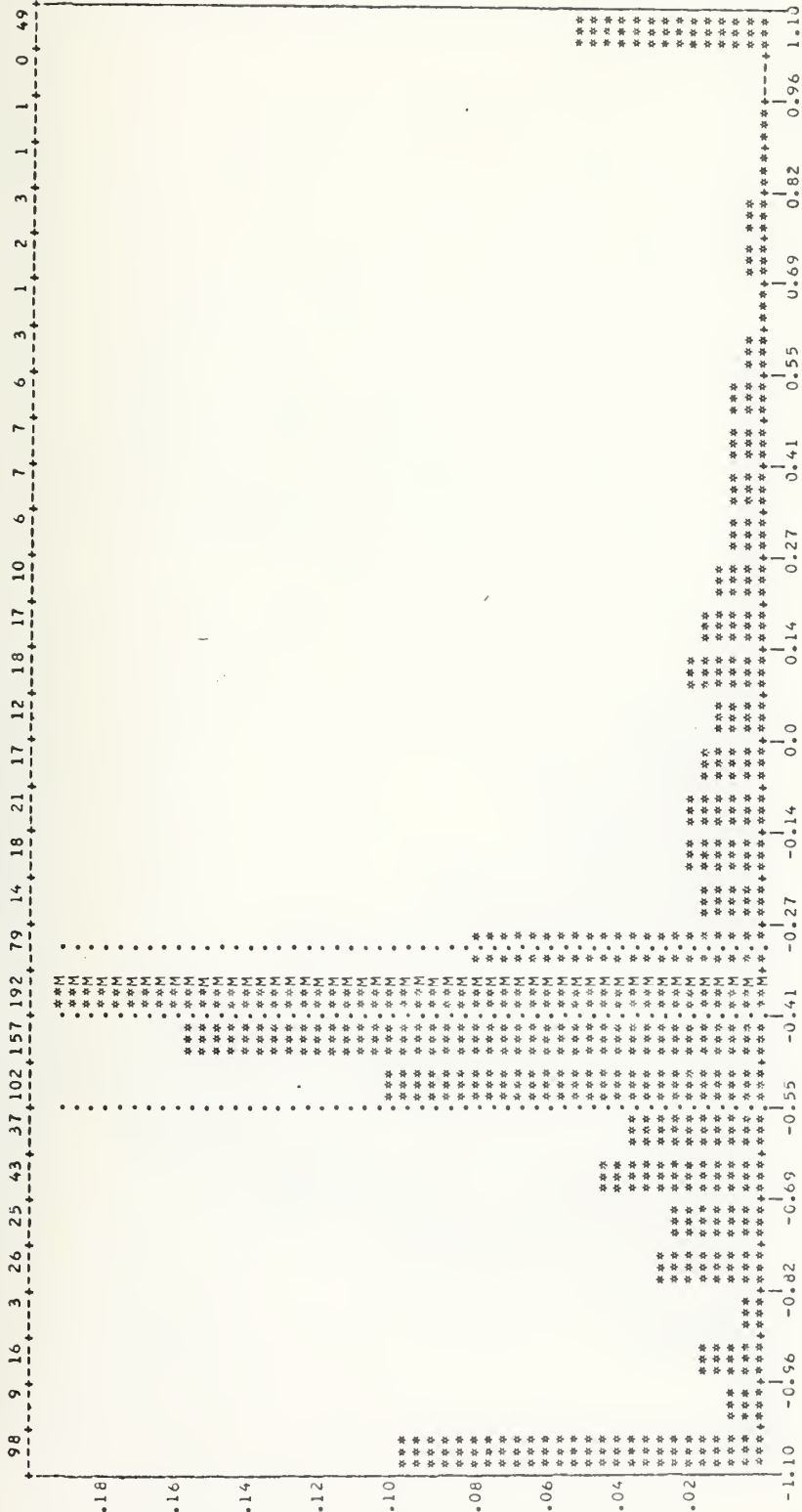


CENTRAL TENDENCY		SPREAD		HIGHER CENTRAL MOMENTS		DISTRIBUTION	
MEAN	-4.25499E 00	VARIANCE	1.048636E 01	M3	-1.463610E 01	MINIMUM	-1.800000E 01
MEDIAN	-4.00000E 00	STD DEV	3.222789E 00	M4	3.833509E 02	10 QUANTILE	-8.000000E 00
TRIMEAN	-4.00000E 00	COEF VAR	7.574121E-01	SKEWNESS	-4.312753E-01	.25 QUANTILE	-6.000000E 00
MIDMEAN	-4.12799E 00	MEAN DEV	2.514959E 00	KURTOSIS	5.99584E-01	.50 QUANTILE (MEDIAN)	-4.000000E 00
MIDRANGE	-7.50000E 00	RANGE	2.100000E 01	BETA1	-1.435282E 01	.75 QUANTILE	-2.000000E 00
		MIDSPREAD	4.000000E 00	BETA2	3.874436E 02	.90 QUANTILE	0.0
						MAXIMUM	3.000000E 00

ERROR IN SIGHT SETTING
 DESIRED DELIVERY
 DIVE ANGLE 45.
 TRUE AIRSPEED 450.
 RELEASE ALTITUDE 4000.

SAMPLE SIZE = 1000

FREQUENCIES



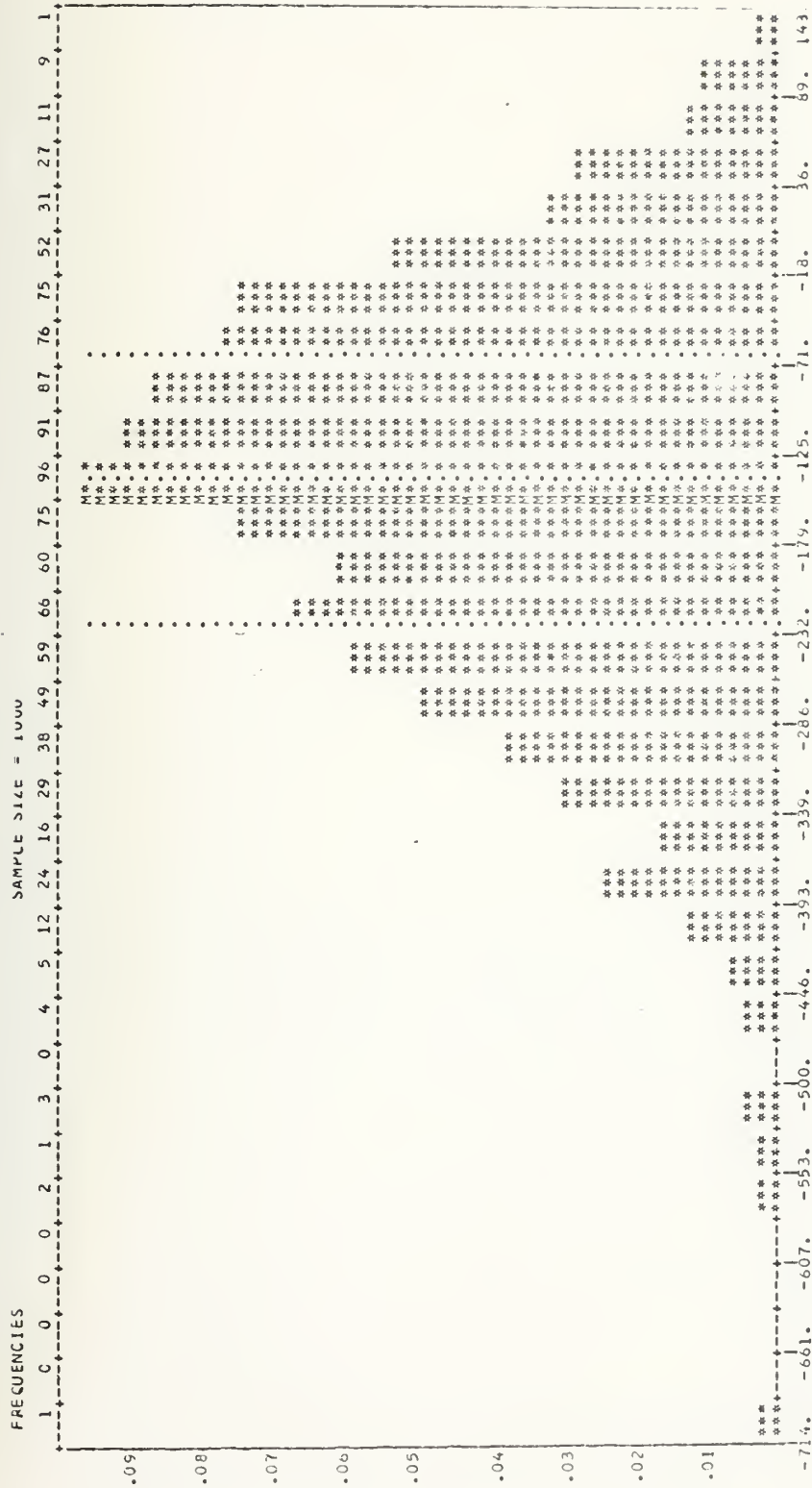
CENTRAL TENDENCY		SPREAD		HIGHER CENTRAL MOMENTS		DISTRIBUTION	
MEAN	-3.712570E-01	VARIANCE	2.320560E-01	M3	1.458954E-01	MINIMUM	-1.099999E 00
MEDIAN	-4.196115E-01	STD DEV	4.817219E-01	M4	2.927083E-01	10 QUANTILE	-8.75779E-01
TRIMEAN	-4.303219E-01	COEF VAR	1.297543E 00	SKENNESS	1.305125E 00	25 QUANTILE	-8.23779E-01
MODE	-4.266163E-01	MEAN DEV	3.037773E-01	KURTOSIS	2.472766E 00	50 QUANTILE	-3.198779E-01
MIDRANGE	0.0	RANGE	2.195999E 00	BETA1	1.454580E-01	75 QUANTILE	1.816664E-01
		MIDSPREAD	2.424904E-01	BETA2	2.938521E-01	90 QUANTILE	1.099999E 00
						MAXIMUM	

CORRECTED ERROR/UNCORRECTED ERROR

DESIRED DELIVERY

LIVE ANGLE 45. RELEASE ALTITUDE 4000.

TRUE AIRSPEED 450.



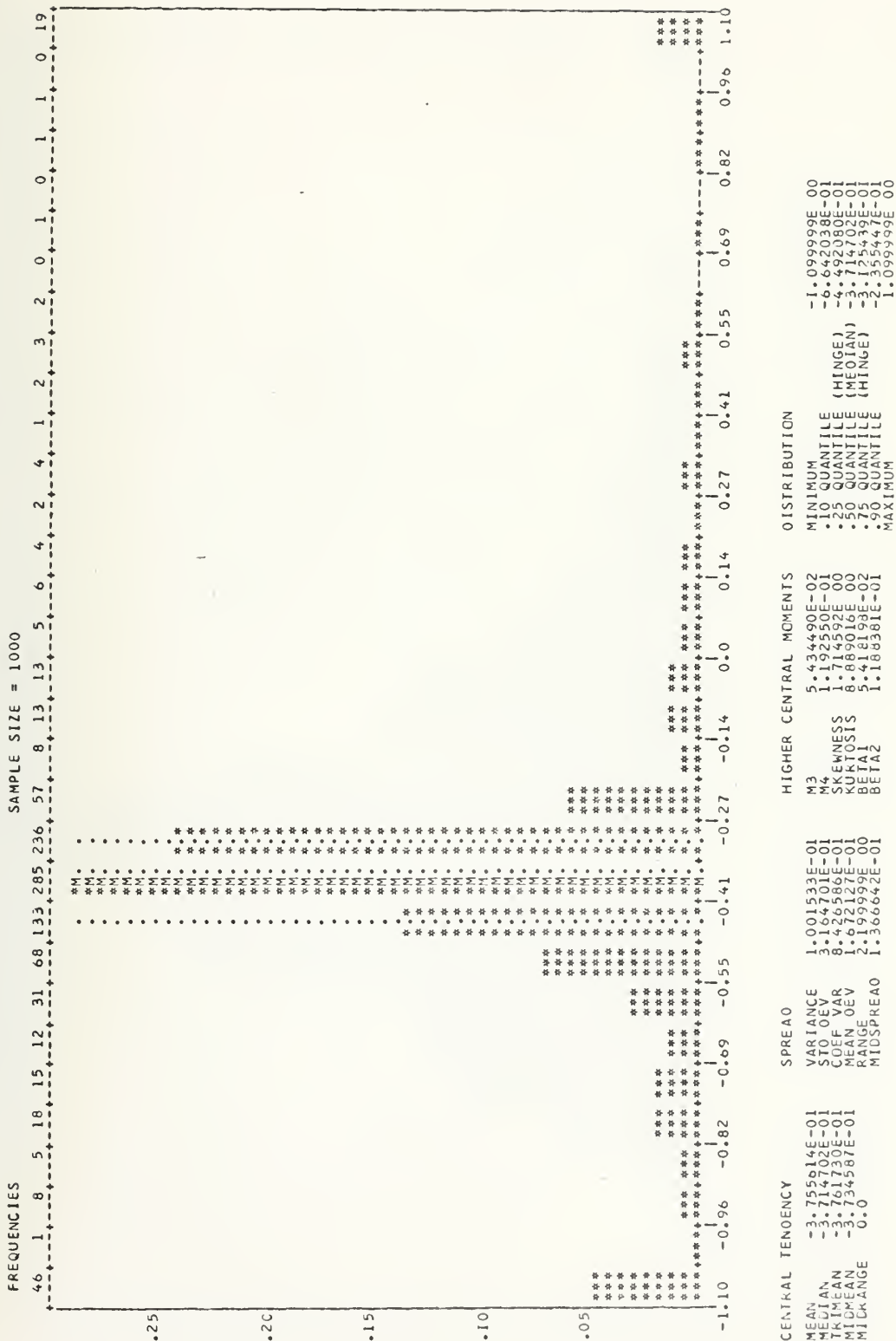
CENTRAL TENDENCY		SPREAD		HIGHER CENTRAL MOMENTS		DISTRIBUTION	
MEAN	-1.487407E 02	VARIANCE	1.473603E 04	M3	-8.503251E 08	MINIMUM	-7.140742E 02
MEDIAN	-1.380823E 02	STDEV	3.838718E 01	M4	8.905979E -01	10 QUANTILE	-3.092471E 02
MODE	-1.397101E 02	COEF VAR	9.559491E -01	SKENNESS	3.023341E -01	25 QUANTILE	-2.59609E 02
MEAN DEV	-2.856897E 02	RANGE	8.587690E 02	KURTOSIS	-8.483842E 08	50 QUANTILE	-1.336855E 02
MIDRANGE		MIDSPREAD	1.632813E 02	BETA1	6.867610E 08	75 QUANTILE	-6.542909E 01
				BETA2		90 QUANTILE	1.426948E 02

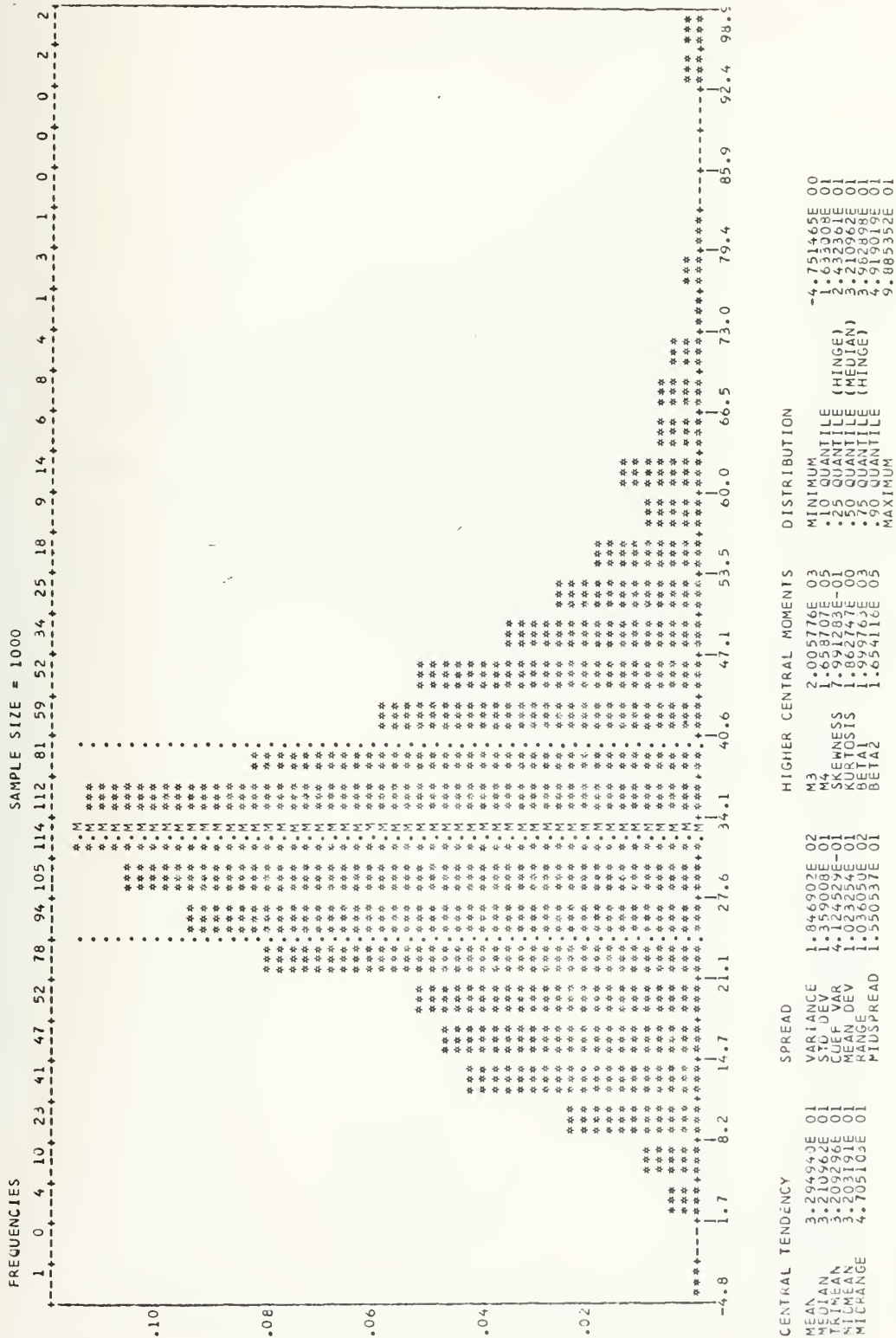
ERROR IN UNCORRECTED DOWN RANGE TRAVEL

DESIRED DELIVERY

DIVE ANGLE TRUE AIRSPEED RELEASE ALTITUDE

45. 500. 6000.





CENTRAL TENDENCY		SPREAD		HIGHER CENTRAL MOMENTS		DISTRIBUTION	
MEAN	3.28494E 01	VARIANCE	1.84690E 02	M3	2.00377E 03	MINIMUM	-4.71465E 00
STANDARD DEVIATION	3.25162E 01	STANDARD DEVIATION	4.30292E 01	M4	1.83591E 05	QUANTILE	1.63308E 01
TRIMED MEAN	3.25031E 01	CUEF VAR	1.03325E 01	SKENESS	1.82347E 00	QUANTILE	1.63308E 01
MICRANGE	4.70510E 01	MEAN DEV	1.03305E 02	KURTOSIS	1.99876E 03	QUANTILE	1.63308E 01
		RANGE	1.55053E 01	BET42	1.65411E 05	QUANTILE	1.63308E 01
		PLUSPREAD				MAXIMUM	9.88535E 01

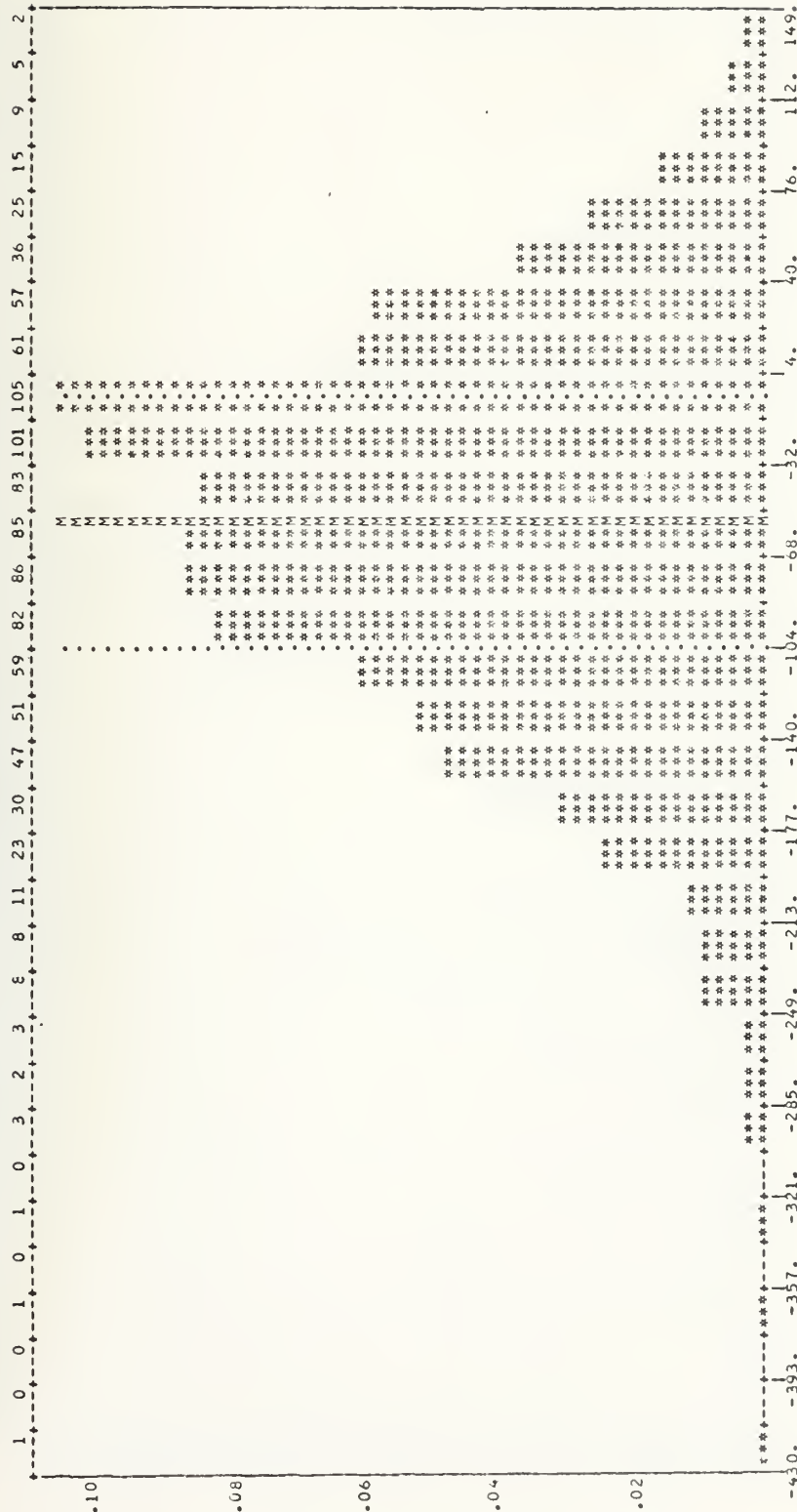
ERROR IN CCWN RANGE TRAVEL

DESIRIED DELIVERY

UTIVE ANGLE 60. TRUE AIRSPEED 500. RELEASE ALTITUDE 7000.

SAMPLE SIZE = 1000

FREQUENCIES



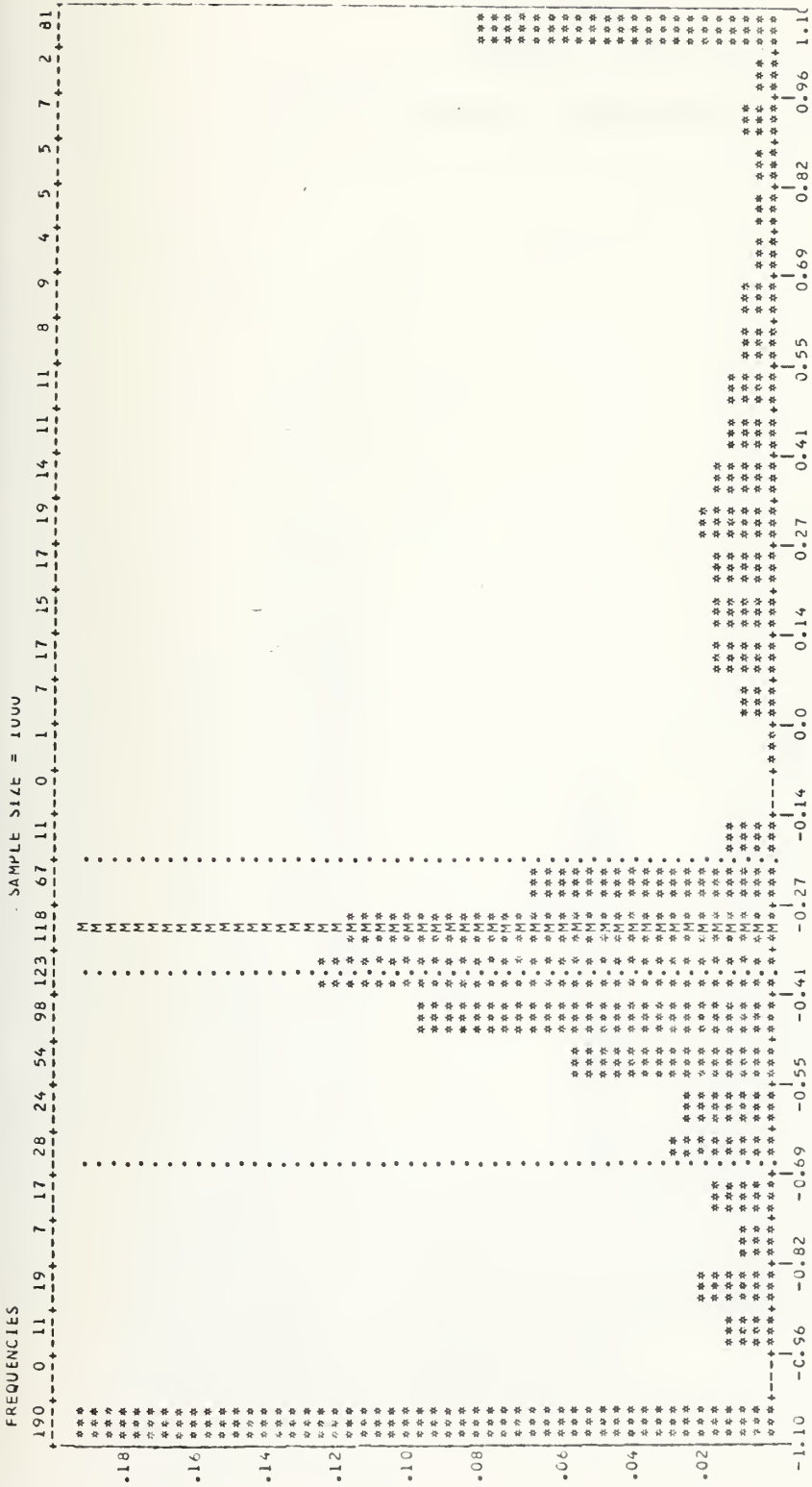
CENTRAL TENDENCY		SPREAD		HIGHER CENTRAL MOMENTS		DISTRIBUTION	
MEAN	-5.545532E 01	VARIANCE	5.885574E 03	M3	-2.073315E 05	MINIMUM	-4.295000E 02
MEOLAN	-5.126025E 01	STD DEV	7.671750E 01	M4	1.270988E 08	.10 QUANTILE	-1.553965E 02
TRIMEAN	-5.230530E 01	COEF VAR	1.383410E 00	SKENESS	-4.591790E -01	.25 QUANTILE	-1.036487E 02
MIDMEAN	-5.071989E 01	MEAN DEV	6.085938E 01	KURTOSIS	6.691370E -01	.50 QUANTILE (MEOTAN)	-5.126025E 01
MIDRANGE	-1.404350E 02	RANGE	5.781221E 02	BETA1	-2.067100E 05	.75 QUANTILE	-3.082000E 00
		MIDSPREAD	1.005961E 02	BETA2	1.2267980E 08	.90 QUANTILE	3.689749E 01
						MAXIMUM	1.486221E 02

ERFOR IN UNCORRECTED CCWN FANGE TRAVEL

DESIRED DELIVERY

DIVE ANGLE TRUE AIRSPEED RELEASE ALTITUDE

60. 500. 7000.



CENTRAL TENDENCY

MEAN -3.086175E-01
MEDIAN -3.810638E-01
TRIMEAN -4.133854E-01
MODE -3.547231E-01
MIDKANGE 0.0

SPREAD

VARIANCE 3.896763E-01
STD DEV 6.242406E-01
COEF VAR 2.022699E 00
MEAN DEV 4.447933E-01
RANGE 2.199999E 00
MIDSPREAD 4.643247E-01

HIGHER CENTRAL MOMENTS

M3 1.984040E-01
M4 4.715572E-01
M5 8.156320E-01
SKURTOSIS 1.081699E-01
BETA1 1.978092E-01
BETA2 4.705889E-01

DISTRIBUTION

MINIMUM -1.099999E 00
10 QUANTILE -6.778698E-01
25 QUANTILE -3.910638E-01
50 QUANTILE (MEDIAN) -2.135451E-01
75 QUANTILE 7.274188E-01
MAXIMUM 1.099999E 00

CORRECTED ERROR/UNCORRECTED ERROR

DESIRED DELIVERY

DIVE ANGLE 60.
TRUE AIRSPEED 500.
RELEASE ALTITUDE 7000.

APPENDIX C

STATIC ACCURACY SIMULATION PROGRAM

```

// EXEC FORTCLG, REGION.GO=100K
// FORT. SYSIN DD *
REAL*4 DX(2)/-200.,200./,DELT(2)/-5.,5./
SINE(X)=SIN(X*.0174533)
COT(X)=COTAN(X*.0174533)
ATAN1(X)=ATAN(X)/.0174533
1  FORMAT(4F10.0)
10  FORMAT(1H0,T33,'TABLE 1 SIMULATION OF DTNET AND DX',)
11  FORMAT(1H0,T30,'DESIRED DIVE',T45,'ACTUAL DIVE',T60,'DX',
1  T65,'CORRECTION ALT',T85,'DTNET',T105,'DX',)
12  FORMAT(1X,T30,F10.5,T40,F10.5,T55,F7.1,T65,F10.2,T80,E15.7,
1  T100,E15.7)
13  FORMAT(1X,' ')
9  FWRITE(6,9)
DO 20 I=1,15
20  WRITE(6,13)
DO 100 I=1,6
READ(5,1) THET,STEEP,SHAL,RELALT
CCRALT=760.5*SINE(THET)*3.+RELALT
DO 50 J=1,2
ATHET=THET+DELT(J)
DO 50 K=1,2
X=CCRALT*COT(ATHET)
DTNET=ATAN1(CORALT/(X-DX(K)))-ATHET
IF ( DTNET.GT. 0.0) GO TO 40
DXP=DTNET*SHAL
WRITE(6,12) THET,ATHET,DX(K),CORALT,DTNET,DXP
GO TO 50
DXP=DTNET*STEEP
WRITE(6,12) THET,ATHET,DX(K),CORALT,DTNET,DXP
40  CONTINUE
50  CONTINUE
100 DO 110 I=1,10
110  WRITE(6,13)
STOP
END

```


APPENDIX D

SIMULATION OF ANGLE OF ATTACK MODULE

```

BEGIN
  INTEGER ARRAY CO, DCO, B(0::21);
  FOR I:=0 UNTIL 21 DO
    READ(CO(I), DCO(I), B(I));
  BEGIN
    INTEGER PROCEDURE MIL(INTEGER VALUE CAS, THET);
    BEGIN
      INTEGER PROCEDURE T(INTEGER VALUE CAS);
      BEGIN
        NUMBER(#FFFF AND BITSTRING( NUMBER(#7C) + NUMBER(BITSTRING(CAS* (
          NUMBER( BITSTRING( NUMBER(#7FD) - NUMBER( BITSTRING (CAS*
            NUMBER(#53)) SHR 4 )) SHR 2)))SHR 7)) SHR 3)
        END T;
      INTEGER PROCEDURE ICOS(INTEGER VALUE THET);
      BEGIN
        CO(NUMBER( BITSTRING(THET) SHR 2)) -NUMBER( BITSTRING( NUMBER
          (BITSTRING( THET) AND #3) * DCO( NUMBER( BITSTRING(THET) SHR 2)))
          SHR 1)
        END ICOS;
        NUMBER( BITSTRING(ICOS(THET) *( B(NUMBER(BITSTRING(THET) SHR 4)) -
          T(CAS))) SHR 9;
      END MIL;
      WRITE("DIVE", "CAS", "MIL");
      FOR I:= 0 STEP 10 UNTIL 60 DO
        FOR J:= 5 STEP 10 UNTIL 255 DO
          WRITE(I, J+295, MIL(J, I));
        END
      END.
    END
  END.

```


APPENDIX E

FORECASTING SIMULATION

```

COMMON /POLY/NTERMS,BI(20),CI(20),DI(20)
COMMON B(6,3),THET,TRST,SLUG,W
REAL*4 WI(100)/100*1.0/
DIMENSION XS(3),YX(3),XA(3),YA(3)
DIMENSION X(100),Y(100),ERROR(100),PJM(100),PJ(100)
DIMENSION YS(3)
DIMENSION POY(2),YAP(2),YSP(2)
DIMENSION ERROR(1000),ERAV(1000),ERSEC(1000)
DIMENSION ERRS(4),DEVS(4)
DIMENSION XMEAN(1000),YN
DIMENSION XMEAN(1000),YMEAN(1000),YCN(1000),XCON(1000)
DIMENSION HT(42),SPD(42)
DIMENSION ZX(4)
DATA DEVS/5000.,25.,3.,1500./
DEFINITION OF TERMS
THET=DIVE ANGLE
TRST = THRUST
SLUG= MASS
W = WEIGHT
B = AN ARRAY OF COEFFICIENTS TO BE USED BY DRAG
NTERMS,BI,CI,DI COEFFICIENTS TO BE USED BY
THE ROUTINE WHICH GENERATES ORTHOGONAL POLYNOMIALS
XS,YS ONE SECOND SAMPLES
XA,YA AVERAGED SAMPLES
POY,YAP,YSP SAMPLES TO BE USED BY ORTHOGONAL POLYNOMIAL PREDICTION
SECOND PREDICTIONS BY ORTHOGONAL, AVERAGING, AND ONE
ERROR,ERAV,ERSEC THE ERROR OF THE ORTHOGONAL, AVERAGED, AND ONE SECOND
PREDICTIONS
DEVS STANDARD DEVIATION OF ENTRY CONDITIONS
XMEAN,YMEAN,YCN,XCON ARRAY OF PSEUDO RANDOM
VARIABLES
TRUNC(X)=FLOAT(IFIX(X+.5))
TRUNC4(X)=FLOAT(IFIX(X*10+.5))/10
STATEMENTS TO CONVERT FLOATING POINT TO SIMULATED
FOUR PLACE PRECISION
CCIN(X)=--1.**IFIX(X+.5)
STATEMENT TO GIVE A RANDOM* + OR - 1 IF X IS UNIFORM (0,1)
101 FORMAT(E15.7)
CALL OVFLOW
REALT=3000.
NPPOINT=20
NTERMS=3
W=420009
W=42000
SLUG=W/32.17
THET=30.
ALT=6000.

```



```

DO 2 I=1,6
DO 1 J=1,3
READ(5,101)B(I,J)
READ INITIAL VALUES FOR DRAG COEFFICIENTS
1 CONTINUE
2 CONTINUE
NNN=1000
TRST=20000.
SEED FOR LLRANDOM
IX=7365247
HT(1)=6000.
SPD(1)=345.
GET RANDOM VARIABLES
CALL SNORM(IX,YMEAN,NNN)
CALL SRAND(IX,YCN,NNN)
CALL SRAND(IX,XCON,NNN)
CALL SNORM(IX,XMEAN,NNN)
VEL=345.
ALIT=6000.
TAKE THE INITIAL SAMPLES
DO 10 I=2,42
CALL INTEG(VEL,.1,ALIT)
SPD(I)=VEL
HT(I)=ALIT
10 CONTINUE
VEL=SPD(42)
ALIT=HT(42)
DELT=.5
DO 20 K=1,20
SPL=ALIT
SPL=VEL
CALL INTEG(VEL,DELT,ALIT)
SIMULATE THE DIVE TO RELEASE ALTITUDE
IF (ABS(ALIT-RELALT) .LE. 10.) GO TO 25
IF (ALIT .GT. RELALT) GO TO 20
DELT=DELT*(ALIT-RELALT)/(ALIT-ALIT)
ALIT=ALITL
VEL=SPL
20 CONTINUE
25 CONTINUE
DO 70 I=1,1000
DO 60 K=1,42
INTRODUCER INSTRUMENT ERRORS
X(K)=TRUNC(HT(K)+COIN(XCON(I))*XMEAN(I))*10.)
Y(K)=TRUNC4(SPD(K)+YMEAN(I)*COIN(YCN(I))*5)
60 CONTINUE
DO 50 J=1,3
ONE SECOND SAMPLES

```



```

XS(J)=X(1+(J-1)*10)
YS(J)=Y(1+(J-1)*10)
50 CONTINUE
GENERATE ORTHOGONAL POLYNOMIAL
CALL ORPOL(NPOINT,X,Y,WI,ERROR,PJM,PJ)
EVALUATE ORTHOGONAL POLYNOMIAL
PRDO=ORTVL(RELALT)
MAKE PREDICTION BASED ON ONE SECOND SAMPLING
CALL INTERP(XS,YS,RELALT,YPS)
ERURT(I)=PRDO-VEL
ERSEC(I)=YPS-VEL
DO 55 J=1,3
XAV=0.0
YAV=0.0
TAKE AVERAGED SAMPLES
DO 53 K=1,4
XAV=XAV+X(K+(9*(J-1)))
YAV=YAV+Y(K+(9*(J-1)))
53 CONTINUE
XA(J)=XAV/4.
YA(J)=YAV/4.
55 CONTINUE
MAKE A PREDICTION FROM AVERAGED SAMPLES
CALL INTERP(XA,YA,RELALT,YPA)
EKAV(I)=YPA-VEL
70 CONTINUE
PLOT THE DISTRIBUTIONS OF THE ERRORS
STOP
END
CALL HISTG(ERAV,NNN,0)
CALL HISTG(ERURT,NNN,0)
CALL HISTG(ERSEC,NNN,0)
SUBROUTINE ORPOL(NPOINT,X,F,W,ERROR,PJM1,PJ)
COMMON /POLY/NTERMS,BI(20),CI(20),DI(20)
DIMENSION X(100),F(100),W(100),ERROR(100),PJML(100),PJ(100)
A SUBROUTINE TO GENERATE A SET OF ORTHOGONAL
POLYNOMIALS OF THE FORM  $P(X)=DI(1)*X^0+DI(2)*X^1+\dots$ 
WHERE  $PO=1$  AND  $PJ+1=(X-BI(J+1))*PJ-C(J+1)*PJ-1$ 
REFERENCE ELEMENTARY NUMERICAL ANALYSIS BY CONTE AND DE BOOR
DO 9 J=1,NTERMS
BI(J)=0.
DI(J)=0.
9 S(J)=0.
CI(1)=0.
DO 10 I=1,NPOINT
DI(I)=CI(I)+F(I)*W(I)
BI(1)=BI(1)+W(I)*X(I)

```

CCC


```

10 S(I)=S(I)+W(I)
   DI(I)=DI(I)/S(I)
   DO 11 I=1,NPOINT
11  ERROR(I)=F(I)-DI(I)
   IF (NTERMS.EQ. 1)
   BI(I)=BI(I)/S(I)
   DO 12 I=1,NPOINT
12  PJM(I)=1.
   PJ(I)=X(I)-BI(I)
20  J=J+1
   DO 21 I=1,NPOINT
   P=PJ(I)*W(I)
   DI(J)=DI(J)+ERROR(I)*P
   P=P*PJ(I)
   BI(J)=BI(J)+X(I)*P
21  S(J)=S(J)+P
   DI(J)=DI(J)/S(J)
   DO 22 I=1,NPOINT
22  ERROR(I)=ERROR(I)-DI(J)*PJ(I)
   IF (J.EQ. NTERMS)
   BI(J)=BI(J)/S(J)
   CI(J)=S(J)/S(J-1)
   DO 27 I=1,NPOINT
   P=PJ(I)
   PJ(I)=(X(I)-BI(J))*PJ(I)-CI(J)*PJM(I)
27  PJM(I)=P
      GO TO 20

      END FUNCTION ORTVL(X)
      COMMON /POLY/NTERMS,BI(20),CI(20),DI(20)
      A FUNCTION TO EVALUATE THE ORTHOGONAL POLYNOMIALS
      GENERATED BY ORTPOL. REQUIRES COMMON LABELED 'POLY'
      REFERENCE ELEMENTARY NUMERICAL ANALYSIS CONTE AND DE BOOR
      K=NTERMS
      ORTVL=DI(K)
      PREV=0.
10  K=K-1
      IF (K.EQ. 0)
      PREV=PREV
      ORTVL=DI(K)+(X-BI(K))*PREV-CI(K+1)*PREV2
      GO TO 10
      END
      SUBROUTINE INTERP(XI,YI,X,PY)
      DIMENSION XI(3),YI(3)
      F10=(YI(2)-YI(1))/(XI(2)-XI(1))
      F21=(YI(3)-YI(2))/(XI(3)-XI(2))

```

C
C
C


```

F210=(F21-F10)/(XI(3)-XI(1))
PY=YI(1)+(X-XI(1))*(F10+(X-XI(2))*F210)
RETURN
END
SUBROUTINE INTEG(V,DT,XHT)
COMMON B(6,3),THET,TRST,SLUG,W
F1(X)=FUN1(X)
F2(X,Y)=X*Y*1.689
X1=DT*F1(V)
Y1=F2(DT,V)
X2=DT*F1(V+X1*.5)
Y2=F2(DT,V+X1*.5)
X3=DT*F1(V+X2*.5)
Y3=F2(DT,V+X2*.5)
X4=DT*F1(V+X3)
Y4=F2(DT,V+X3)
V=V+.1666667*(X1+2.*X2+2.*X3+X4)/1.689
XHT=XHT-SIN(THET*.017453)*(Y1+2.*Y2+2.*Y3+Y4)*.1666667
RETURN
END
FUNCTION FUN1(X)
COMMON B(6,3),THET,TRST,SLUG,W
FUN1=(TRST+SIN(THET*.017453)*W-DRAG(X))/SLUG
RETURN
END
FUNCTION DRAG(TAS)
COMMON B(6,3),THET,TRST,SLUG,W
FX(B1,B2,B3,X)=B1+X*(B2+X*B3)
IF(THET-10.)10,20,40
D20=FX(B(5,1),B(5,2),B(5,3),TAS)
D10=FX(B(6,1),B(6,2),B(6,3),TAS)
DRAG=(D10-D20)*(THET-20.)*(-.1)+D20
RETURN
20 DRAG=FX(B(6,1),B(6,2),B(6,3),TAS)
RETURN
40 IF(THET-20.)10,50,70
50 DRAG=FX(B(5,1),B(5,2),B(5,3),TAS)
RETURN
60 D20=FX(B(5,1),B(5,2),B(5,3),TAS)
D30=FX(B(4,1),B(4,2),B(4,3),TAS)
DRAG=(D20-D30)*(THET-30.)*(-.1)+D30
RETURN
70 IF(THET-30.)60,80,100
80 DRAG=FX(B(4,1),B(4,2),B(4,3),TAS)
RETURN
90 D45=FX(B(3,1),B(3,2),B(3,3),TAS)
D30=FX(B(4,1),B(4,2),B(4,3),TAS)
DRAG=(D30-D45)*(THET-45.)*(-.1/15.)+D45

```



```

100 RETURN
110 IF (THET-45.) 90,110,130
    DRAG=FX(B(3,1),B(3,2),B(3,3),TAS)
    RETURN
120 D45=FX(B(3,1),B(3,2),B(3,3),TAS)
    D50=FX(B(2,1),B(2,2),B(2,3),TAS)
    DRAG=(D45-D50)*(-1./15.)+D50
    RETURN
130 IF (THET-50.) 120,140,160
140 DRAG=FX(B(2,1),B(2,2),B(2,3),TAS)
    RETURN
150 D60=FX(B(1,1),B(1,2),B(1,3),TAS)
    D50=FX(B(2,1),B(2,2),B(2,3),TAS)
    DRAG=(D50-D60)*(-.1)+D60
    RETURN
160 IF (THET-60.)150,170,150
170 DRAG=FX(B(1,1),B(1,2),B(1,3),TAS)
    RETURN
    END

```


APPENDIX F

DELIVERY ACCURACY COMPARISON

```

1 // EXEC FORTCLG, REGION.GO=120K
2 //FORT
3 DIMENSION EREN(4000),ZX(4),CX(4),EREAD(2),CRX(2),ERROR(10)
4 1,EMILS(1000),EDRT(1000),EDRTU(1000),PER(1000)
5 1,COMMON B(6,3),THET,TRST,SLUG,W
6 LOGICAL FLAG
7 DATA VE/7.2/
8 DATA NN/1000/
9 DATA ZX/200.,10.,2.5,750./
10 DATA XMAL,XMAS,XMDI,XMTR/6000.,350.,28.5,18000./
11 COIN(X)=-1.**(IFIX(X+.5))
12 TRUNC(X)=FLOAT(IFIX(X*10+.5))/10.
13 FFORMAT(EI5.7)
14 FFORMAT(I7)
15 FFORMAT(F10.0)
16 FFORMAT(F10.0)
17 FFORMAT(I10,I10,'DESIRED DELIVERY',//
18 11X,'DIVE ANGLE',T20,'TRUE AIRSPEED',T40,'RELEASE ',
19 2,ALTITUDE,//1X,F10.0,T20,F10.0,T45,F10.0)
20 FFORMAT(1X,'ERROR IN SIGHT SETTING')
21 FFORMAT(1X,'ERROR IN DOWN RANGE TRAVEL')
22 FFORMAT(1X,'ERROR IN UNCORRECTED DOWN RANGE TRAVEL')
23 FFORMAT(1X,'CORRECTED ERROR/UNCORRECTED ERROR')
24 CALL OVFLOW
25 W=42000
26 SLUG=1305.56
27 READ(5,1)((B(I,J),J=1,3),I=1,6)
28 READ(5,2)IX
29 DO 310 I=1,6
30 READ(5,3)(ERROR(I),I=1,10)
31 READ(5,4)XMAL,XMAS,XMDI,XMTR
32 CALL SNORM(IX,EREN,4000)
33 DO 300 KK=1,NN
34 CALL RESET
35 CALL SRAND(IX,CX,4)
36 ALT=XMAL+EREN(KK)*ZX(1)*COIN(CX(1))
37 VEL=XMAL+EREN(KK+NN)*ZX(2)*COIN(CX(2))
38 THET=XMDI+EREN(KK+2*NN)*ZX(3)*COIN(CX(3))
39 TRST=XMTR+EREN(KK+3*NN)*ZX(4)*COIN(CX(4))
40 DT=1.0
41 CAS=VEL*((-1407E-04*ALT)+1.0)+5.0
42 CALL SAMP(VEL,CAS,THET,FLAG,ALT,.FALSE.,MIL,MILS,
43 1ERROR)
44 CALL SNORM(IX,EREAD,2)
45 CALL SRAND(IX,CRX,2)
46 DO 50 K=1,30
47 SPL=VEL

```



```

1  ALTL=ALTI
   CALL INTEG(VEL,DT,ALT)
   IF (ALT.LT.(ERROR(9)-10.)) GO TO 40
   CAS=VEL*((-1.407E-04*ALT)+1.)+5.
   TAS=TRUNC4(VEL+COIN(CRX(1))*EREAD(1)*.5)
   ALTI=TRUNC(ALT+COIN(CRX(2))*EREAD(2)*10.)
   CALL SAMP(TAS,CAS,THET,FLAG,ALTI,.TRUE.,MIL,
   MILS,ERROR)
   IF (FLAG) CALL ADTHET(TAS,THET,THETO,ALT,MIL)
   IF (ABS(ALT-ERROR(9)) .LE. 10.) GO TO 60.
   GO TO 50
40  DT=DT*((ALTL-ERROR(9))/(ALTL-ALT))
   ALTI=ALTL
   VEL=SPL
   CONTINUE
50  ALTR=VEL
60  VELR=VEL
   ALTR=VEL
   THETA=THET +.0572958*FLOAT(MIL)
   THET=THETA
   CALL TRAJ(VEL,THETA,VE,ALT,DRT)
   MILSC=MIL+FIX(1000.*(ATAN(ALTR/DRT)-THETA*.0174533))
   EMILS(KK)=FLOAT(MILS-MILSC)
   DRTC=ALTR/TAN(.0174533*THETA+.001*FLOAT(MILS-MIL))
   EDRT(KK)=DRTC-DRT
   THET=THETO
   CALL TRAJ(VELR,THET,VE,ALTR,DRTU)
   DRTA=ERROR(9)/TAN(THET*.0174533+.001*(ERROR(10)-
   FLOAT(MIL)))
   EDRTU(KK)=DRTU-DRTA
   PER(KK)=EDRT(KK)/EDRTU(KK)
   IF (PER(KK) .LT. -1.0) PER(KK)=-1.1
   IF (PER(KK) .GT. 1.0) PER(KK)=1.1
300 CONTINUE
   CALL HISTG(EMILS,NN,0)
   WRITE(6,101)
   WRITE(6,5)(ERROR(I),I=7,9)
   CALL HISTG(EDRT,NN,0)
   WRITE(6,102)
   WRITE(6,5)(ERROR(I),I=7,9)
   CALL HISTG(EDRTU,NN,0)
   WRITE(6,103)
   WRITE(6,5)(ERROR(I),I=7,9)
   CALL HISTG(PER,NN,0)
   WRITE(6,104)
   WRITE(6,5)(ERROR(I),I=7,9)
310 CONTINUE
   STOP
   END

```



```

SUBROUTINE RESET
I=0
RETURN
ENTRY ADTHET(DX, THET, THETO, ALT, MIL)
IF (I.GT. 0) GO TO 10
THETO=THET
I=1
10 DIVE=THET*.0174533+.001*FLOAT(MIL)
X=ALT/TAN(DIVE)
DIVEN=A*TAN(ALT/(X+DX))
THET=57.2958*(DIVEN-.001*FLOAT(MIL))
RETURN
END
SUBROUTINE TRAJ(TAS, THXT, VE, ALT, ZZ)
DIMENSION VX(2), VY(2), X(2), Y(2)
F1(Y,V,VX)=-DRIG(V,Y)*VX
F2(Y,V,VY)=-DRIG(V,Y)*VY-32.17
DT=1.0
XM=260./32.17
VAC=TAS*.689
VB=SQR(VE**2+VAC**2)
THET=THXT*.01745
PHI=THET+ATAN(VE/VAC)
VX(1)=VB*COS(PHI)
VY(1)=-VB*SIN(PHI)
X(1)=ALT+18.*SIN(THET)-7.*COS(THET)
Y(1)=-18.*COS(THET)+7.*SIN(THET)
10 CONTINUE
VX1=F1(Y(1),VB,VX(1)) *DT
VY1=F2(Y(1),VB,VY(1)) *DT
X1=VX(1) *DT
Y1=VY(1) *DT
VEL=SQR((VX(1)+.5*VX1)**2+(VY(1)+.5*VY1)**2)
YT=Y(1)
VX2=DT*F1(YT,VEL,.5*VX1+VX(1))
VY2=DT*F2(YT,VEL,.5*VY1+VY(1))
X2=DT*(VX(1)+.5*VX1)
Y2=DT*(VY(1)+.5*VY1)
VEL=SQR((VX(1)+.5*VX1)**2+(VY(1)+.5*VY1)**2)
VX3=DT*F1(YT,VEL,VX(1)+.5*VX2)
VY3=DT*F2(YT,VEL,VY(1)+.5*VY2)
X3=DT*(VX(1)+.5*VX2)
Y3=DT*(VY(1)+.5*VY2)
VEL=SQR((VX(1)+.5*VX2)**2+(VY(1)+.5*VY2)**2)
VX4=DT*F1(YT,VEL,VX(1)+.5*VX3)
VY4=DT*F2(YT,VEL,VY(1)+.5*VY3)
X4=DT*(VX(1)+.5*VX3)
Y4=DT*(VY(1)+.5*VY3)

```



```

X(2)=X(1)+.166667*(X1+2.*X2+2.*X3+X4)
Y(2)=Y(1)+.166667*(Y1+2.*Y2+2.*Y3+Y4)
VY(2)=VY(1)+.1666667*(VY1+2.*VY2+2.*VY3+VY4)
VX(2)=VX(1)+.1666667*(VX1+2.*VX2+2.*VX3+VX4)
IF(Y(2))40,50,60
IF(Y(2).GT.-5.)GO TO 50
DT=DT*Y(1)/(Y(1)-Y(2))
GO TO 10
CONTINUE
ZZ=X(2)
RETURN
IF(Y(2).LT.5.)GO TO 50
TIME=TIME+DT
VX(1)=VX(2)
X(1)=X(2)
VY(1)=VY(2)
Y(1)=Y(2)
VB=SQR( VY(1)**2+VX(1)**2)
GO TO 10
END
FUNCTION DRIG(VB,Y)
DIMENSION CD(7),XMN(7),VS(7)
DATA CD/.0805,.0805,.0872,.0991,.1421,.2327,.2446/
DATA XMN/.07,810,870,90,960,1,05,1,09/
DATA VS/661.7,659.5,657.2,654.9,652.6,650.3,647.9/
AY=Y/1000
I=IFIX(AY)
X=AY-FLOAT(I)
ASS=(VS(I+1)-VS(I))*X+VS(I+1)
AMN=VB/(ASS*1.689)
I=2
IF(AMN.LT.XMN(I))GO TO 20
I=I+1
IF(I.GT.7)GO TO 15
GO TO 10
WRITE(6,101)
FORMAT(IH0,'XMN TOO BIG')
DRIG=0.0
RETURN
AKD=CD(I-1)+((CD(I)-CD(I-1))/(XMN(I)-XMN(I-1)))*(AMN-XMN(I-1))
DRIG=1.62555E-04*EXP(.00003158*Y)*VB*AKD
RETURN
END

```



```

SUBROUTINE SAMP(TS,CAS,THAT,SWIT,HT,SHFT,MIL,MILS,ERRAR)
DIMENSION ERRAR(10)
DIMENSION ERROR(10),ALT(3),TAS(3),FX(2)
DATA T1/4.0/
LOGICAL SWIT
LOGICAL SHFT
T(X)=-164.415 + X*(.8121744-X*.6827249E-03)
B(X)=79.67641 + X*(.1399574-X*.7575896E-02)
MIL=-COS(THAT*.0174533)*(T(CAS)-B(THAT))
DX=0.0
SWIT=.FALSE.
THET=THAT+FLOAT(MIL)*.057295
IF(SHFT) GO TO 30
DO 1 I=1,10
  ERROR(I)=ERRAR(I)
  ALT1=SIN(ERROR(7)*.01745)*ERROR(8)*1.689*T1+ERROR(9)
  DIVL=ERROR(7)
  SPED=ERROR(8)
  K=0
  STIP=ERROR(3)/ERROR(6)
  SHAL=ERROR(4)/ERROR(5)
  FAST=ERROR(1)/ERROR(6)
  SLOW=ERROR(2)/ERROR(5)
  MILS=IFIX(ERROR(10))
  NIL=MIL
  RETURN
CCONTINUE
THAT=THAT+FLOAT(NIL-MIL)*.0572958
NIL=MIL
IF(HT.LT.ALT1) RETURN
K=K+1
IF(K.EQ.3) MILS=IFIX(ERROR(10))
IF(THET-DIVL) 10,100,20
MILS=MILS+(DIVL-THET)*SHAL
DX=ERROR(4)*(DIVL-THET)
GO TO 100
MILS=MILS+(DIVL-THET)*STIP
DX=ERROR(3)*(DIVL-THET)
DIVL=THET
GO TO (110,120,130),K
110 ALT(1)=HT
TAS(1)=TS
P2=TS
RETURN
120 TAS(2)=TS

```



```

ALT(2)=HT
-FX(2)=(TAS(2)-TAS(1))/(ALT(2)-ALT(1))
P2=P2+FX(2)*(ERROR(9)-ALT(1))
DIVL=ERROR(7)
RETURN
130 P2=P2+(((TS-TAS(2))/(HT-ALT(2))-FX(2))/(HT-ALT(1)))*(ERROR(9)-
    1ALT(1))*(ERROR(9)-ALT(2))
    K=0
    SWIT=TRUE.
    IF(IFIX(P2)-IFIX(SPED))140,160,150
    MILS=MILS+SLOW*(SPED-P2)
    TS=DX+(SPED-P2)*ERROR(2)
    GO TO 160
    150 MILS=MILS+FAST*(SPED-P2)
    TS=DX+(SPED-P2)*ERROR(1)
    160 CONTINUE
    RETURN
END
//GO.FT06F001 DD SYSOUT=D
//GO.SYSIN DD *
-7.32500E00
-5.41608E01
-3.93751E-02
-9.46706E00
-5.33395E01
-3.81394E-02
-1.57772E01
-5.14052E01
-3.44066E-02
-3.14242E01
-4.99344E01
-2.90602E-02
-1.34523E01
-6.46076E01
-5.73131E-02
-9.46034E01
-4.06686E01
-1.03355E-02
-1.547851
23.7
3.3
3.5
4.1
9.7
9.9
20.
450.
2000.

```


23.			
24.			
8.6			
8.6			
50.			
500.			
7000.			
77.	380.	60.	4000.
13000.			

PLM AIR TO GROUND PROGRAM

102


```

THET = THET + SHR( 353Q * MIL(CAS,THET),12);

IF THET > RELDIV THEN /* STEEP */
CM = RELMIL - SHR( STIP *(THET - RELDIV)+4,3);
ELSE /* SHALLOW */
CM = RELMIL + SHR( SHAL *(RELMIL -THET)+4,3);
IF RELDIV = 0 THEN DO;
/* FOR A LEVEL DELIVERY CORRECT FOR DIVE AND
CURRENT AIRSPEED */
IF TAS > RELAS
THEN DO; /* FAST DECREASE SIGHT ANGLE */
CM=CM- SHR( FAST*(TAS-RELAS)+32, 6) ;
RETURN CM;
END;
ELSE DO; /* SLOW ADD TO SIGHT */
CM=CM+SHR( SLOW *(RELAS-TAS) +32, 6 );
RETURN CM;
END;
END; CASE K ;
/* PASS 1 */
DO;
/* X:= RELEASE ALTITUDE */
CALL FLOAT(.RELALT,.X);
/* F0:=TAS; F0:=-F0 */
CALL FLOAT(.TAS,.F0); F0(2)=F0(2) OR 80H;
/* X0:=ALT; X0:=-X0 */
CALL X0:=ALT; X0:=-X0;
/* F0(2)=X0(2) OR 80H;
/* T1:=X+X0 ( REALLY X- ABS(X0) ) */
CALL ADD(.X,.X0,.T1);
A1: RETURN CM ;
END; SECOND PASS */
DO; F10 = (F1 - F0)/(X1 - X0) */
CALL FLOAT(.TAS,.F1); /* F1 := TAS */
CALL FLOAT(.ALT,.X1); /* X1 := ALT */
/* DUM1:= F1 + F0; ( REALLY F1 - ABS(F0) ) */
/* F0:=-F0 */
CALL ADD(.F1,.F0,.DUM1); F0(2) =F0(2) XOR 80H;
A2: /* DUM2 := X1 + X0; ( X1 - ABS (X0 ) ) */
CALL ADD(.X1,.X0,.DUM2);
/* CALCULATE F10 */
CALL DIV(.DUM1,.DUM2,.F10);
A3: /* MF10 := -F10 */

```



```

MF10 = F10;
MF10(1) = F10(1);
MF10(2) = F10(2);
X1(2) = X1(2) OR 80H;
F1(2) = F1(2) OR 80H;
CALL ADD(.X,.X1,.T2);
RETURN CM;
END;

/* PASS 3 */
DO;
CALL FLOAT(.TAS,.F2);
CALL FLOAT(.ALT,.X2);
/* F2 := TAS */
/* X2 := ALT */
A4: /* F21 = (F2 - F1)/(X2 - X1) */
CALL ADD(.F2,.F1,.DUM1);
CALL ADD(.X2,.X1,.DUM2);
CALL DIV(.DUM1,.DUM2,.F21);
/* F210 = (F21 - F10)/(X2 - X1) */
CALL ADD(.F21,.MF10,.DUM1);
CALL ADD(.X2,.X0,.DUM2);
CALL DIV(.DUM1,.DUM2,.F210);
/* FX = F0 + (X-X0)*(F10+F210*(X-X1)) */
CALL MUL(.F210,.T2,.DUM1);
CALL ADD(.DUM1,.F10,.DUM2);
CALL MUL(.DUM2,.T1,.DUM1);
CALL ADD(.DUM1,.F0,.FX);
CALL FIX(.FX,.TAS);
A6:
IF TAS > RELAS
THEN DO; /* PREDICTED FAST DECREASE SIGHT ANGLE */
CM=CM-SHR(FAST*(TAS-RELAS)+32,6);
RETURN CM;
END;
ELSE DO; /* PREDICTED SLOW INCREASE SIGHT ANGLE */
CM=CM+SHR(SLOW*(RELAS-TAS)+32,6);
RETURN CM;
END;
END; /* END OF CASE K */
END MILS;
WEAPON = 0;
LCGP1: DELIV=3;
CALL SET$UP;
LOOP2: CAS=100;

```



```

      THET=30;
      K= MIL(CAS,THET);
      LOOP3: THET= 28;
              TAS=DITA;
              ALT=DITA(1);
              K=0;
              DISP=MILS;
              CALL TIME(PERIOD);
              CALL TIME(PERIOD);
      LOOP4: THET=28;
              TAS=DITA(2);
              ALT= DITA(3);
              K=1;
              DISP=MILS;
              CALL TIME(PERIOD);
              CALL TIME(PERIOD);
      LOOP5: THET=30;
              TAS= DITA(4);
              ALT=DITA(5);
              K=2;
              DISP=MILS;
              HALT;
      EXIT:
      EOF

```


APPENDIX H

PLM AIR TO AIR PROGRAM

```

EXEC PLM80CG, REGION. PASS1=150K, REGION. PASS2=150K
//PASS1.CARD DD *
$SYMBOLS=1
$R=80
$M=1

FLOAT: PROCEDURE(INN,FA);
/* A PROCEDURE TO CONVERT AN INTEGER INN TO
A FLOATING POINT WHOSE MANTISSA IS FOUND IN
ADDRESSES FA AND FA+1 AND WHOSE EXPONENT
IS IN FA +2 */
DECLARE (INN,FA) ADDRESS ,MANN ADDRESS,
MANT BASED FA BYTE,
INT BASED INN ADDRESS,
N BYTE;

IF INT = 0 /* CHECK FOR ZERO INPUT */
THEN DO;
MANT(2),MANT(1),MANT = 0;
RETURN;
END;

MANT(2) = 50H; /* EXPONENT = 2**16 */
MANN = INT;

IF ( MANN AND 8000H) = 8000H THEN /* NEGATIVE INTEGER */
DO; MANT(2) = MANT(2) OR 80H; /* SET SIGN BIT */
MANN = 1-MANN; /* COMPLEMENT */
END;

N = 1; /* NORMALIZE MANTISSA */
DO WHILE (( SHL(MANN,1) AND 8000H) <> 8000H);
N=N+1;
MANN = SHL(MANN,1);
END;

MANN= SHL(MANN,1);
MANT(2) = MANT(2) - N; /* ADJUST EXPONENT */
MANT = HIGH(MANN); /* ASSIGN MANTISSA */
MANT(1) = LOW(MANN);
RETURN;
END FLOAT;

FIX: PROCEDURE(FP,INT); /* REAL TO POSITIVE INTEGER */
DECLARE (FP,INT) ADDRESS, INT ADDRESS,
M BASED FP BYTE, I BASED INT ADDRESS,
N BYTE;

```



```

IF((M(2) AND 80H)=80H) OR ((M(2) AND 40H)=00H)
THEN DO:
  I=.0;
  RETURN
END;

IF( N:= M(2) AND 3FH)> 15 /* EXPONENT TOO LARGE */
THEN DO;
  I=7FFFH; /* RETURN MAXIMUM INTEGER */
  RETURN;
END;

I=M; I=(SHL(I,8) OR M(1)) +SHL(1,15-N);
I = SHR(I,(16-N)); /* RIGHT JUSTIFY */
END FIX;
DECLARE DISP BYTE; DECLARE ZE BYTE;
DECLARE ZZ ADDRESS; DIVIDE ROUTINE */
/* FLOATING POINT DIVIDE ROUTINE */
PROCEDURE (XA,YA,PA);
DECLARE (XA,YA,ZA,B,XX,YQ) ADDRESS,
PA ADDRESS, P BASED PA BYTE,
I BYTE, XA BYTE,
X BASED YA BYTE,
Y BASED YA BYTE,
C (3) BYTE;

XX=SHL(DOUBLE(X),8) OR X(1);
YQ=SHL(DOUBLE(Y),8) OR Y(1);
IF XX=0H THEN DO;
  ZZ=0H; ZE=0H; GO TO RET;
END;
XX=SHR(XX,1);
YC=SHR(YQ,1);
ENTO: ZZ=0;
C(2)= X(2);
I=1;
B=XX-YQ;
IF NOT CARRY THEN DO;
  IF B=0 THEN DO;
    ZZ=8000H;
    ZE=((X(2) AND 80H) XOR (Y(2) AND 80H))
    OR ((C(2) AND 7FH)-(Y(2) AND 7FH)+65));
    GO TO RET;
  END;

```



```

GOTO ENT2;
END;
XX=SHL(X,1);
C(2)=(C(2) AND 7FH) - 1;
ENT1: B=XX-YQ; THEN DO;
      XX=SHL(X,1);
      GOTO ENT3;
END;
/* NO CARRY; ADD ONE TO DIVIDEND TO MANTISSA AND SHIFT LEFT */
ENT2: XX=SHL(B,1);
      ZZ=ZZ+1;
/* CARRY SHIFT DIVIDEND MANTISSA LEFT */
ENT3: ZZ=SHL(ZZ,1);
      I=I+1;
      IF IK 16 THEN GOTO ENT1;
/* SET EXPONENT */
      ZE=((X(2) AND 80H) XOR ((C(2) AND 7FH) - (Y(2) AND 80H))) OR
          ((C(2) AND 7FH) - (Y(2) AND 7FH)+65));
      RET: P=HIGH(ZZ);P(1)=LOW(ZZ);P(2)=ZE;
          RETURN;
END DIV;
/* FLOATING POINT MULTIPLY ROUTINE */
MUL: PROCEDURE(XA,YA,PA);
      DECLARE
        XA ADDRESS, X BASED XA BYTE;
        YA ADDRESS, Y BASED YA BYTE;
        PA ADDRESS, P BASED PA BYTE;
        TQ ADDRESS;
      IF (X=0) OR (Y=0) THEN DO; ZZ=0; ZE=0; GO TO RET; END;
      ZZ=X(1)*Y(1);
      TQ=X(1)*Y(1)+ZZ;
      IF CARRY THEN ZZ=X*Y+256+HIGH(TQ);
      ELSE ZZ=X*Y+HIGH(TQ);
      IF LOW(TQ)>=80H THEN DO; ZZ=SHL(ZZ,1);
      IF ZZ<8000H THEN DO;
        IF LOW(TQ)>=0C0H THEN ZZ=ZZ+1;
        ZE=-1;
      END;
      ELSE ZE=0;
/* ADD EXPONENTS */
      ZE=((X(2) AND 7FH) + (Y(2) AND 7FH)-64+ZE) OR((X(2) AND 80H)
        XOR (Y(2) AND 80H));
      RET: P=HIGH(ZZ);P(1)=LOW(ZZ);P(2)=ZE;
          RETURN;
END MUL;
/* FLOATING POINT ADD ROUTINE */
ADD: PROCEDURE(XA,YA,PA);

```



```

DECLARE (XA,YA,ZA) ADDRESS;
PA ADDRESS, P BASED PA BYTE,
(I,R2,Y2,DIGIT,SINE) BYTE,
X BASED XA BYTE,
Y BASED YA BYTE,
(X,X,YQ) ADDRESS;

/* PROCEDURE TO LEFT JUSTIFY MANTISSA IN BINARY */
ADJUST: PROCEDURE;
DO I=0 TO 15;
  IF (ZZ AND 8000H)=8000H THEN RETURN;
  ZZ=SHL(ZZ,I);
  ZE=ZE-I;
END; RETURN;
END ADJUST;

XX=SHL(DOUBLE(X),8) OR X(1);
YQ=SHL(DOUBLE(Y),8) OR Y(1);
R2= X(2) AND 7FH;
Y2= Y(2) AND 7FH;
DIGIT=R2-Y2;
SINE=(X(2) AND 80H) XOR (Y(2) AND 80H);
IF (DIGIT AND 80H)<>0 THEN DIGIT=-DIGIT;
IF (DIGIT >= 16 THEN DO;
  /* VARIABLES NOT WITHIN SIGNIFICANCE RANGE */
  IF R2 > Y2 THEN DO;
    ZZ=XX; ZE=X(2); GO TO RET;
  END;
  ZZ=YQ; ZE=Y(2); GO TO RET;
END;
IF Y2 = R2 THEN DO;
  /* EXPONENTS EQUAL IN ABSOLUTE VALUE */
  IF YQ>XX THEN DO;
    /* Y > X */
    ZE= Y(2);
    IF SINE < 80H THEN GO TO EXIT1;
    GO TO EXIT2;
  END;
  IF YQ<XX THEN DO;
    /* X > Y */
    ZE =X(2);
    IF SINE < 80H THEN GO TO EXIT1;
    GO TO EXIT3;
  END;
  /* X = Y */
  IF SINE < 80H THEN DO;
    ZE=X(2);
    GO TO EXIT1;
  END;

```



```

ZZ=0;
ZE = 0;
GO TO RET;
END;
IF Y2 > R2 THEN DO;
/* Y > X */
ZE = Y(2);
XX=SHR(XX,DIGIT);
IF SINE < 80H THEN GO TO EXIT1;
GO TO EXIT2;
END;
/* X > Y */
ZE = X(2);
YQ=SHR(YQ,DIGIT);
IF SINE < 80H THEN GO TO EXIT1;
GO TO EXIT3;
EXIT1:
ZZ=XX+YQ; THEN DO;
IF CARRY
ZZ=SCR(ZZ,1);
ZE=ZE +1;
END;
GO TO RET;
EXIT2:
ZZ=YQ-XX;
CALL ADJUST;
GO TO RET;
EXIT3:
ZZ=XX-YQ;
CALL ADJUST;
RET: P=HIGH(ZZ);P(1)= LOW(ZZ);P(2)=ZE;
END ADD;
SUB: PROCEDURE(FIRST,SECOND,THIRD);
DECLARE (FIRST,SECOND,THIRD) ADDRESS, X BASED SECOND
BYTE;
X(2)=X(2) XOR 80H;
CALL ACD(FIRST,SECOND,THIRD);
RETURN;
END SUB;
MULT: PROCEDURE(X,Y,Z);
DECLARE (X,Y,Z) ADDRESS;
CALL MUL(X,Y,Z);
RETURN;
END MULT;
COMPARE: PROCEDURE (XA,YA) BYTE;
/* ROUTINE TO COMPARE TWO FLOATING POINT VARIABLES */

```



```

/* IF X<Y COMPARE=0
   X=Y COMPARE=1
   X>Y COMPARE=2 */
DECLARE (XE,YE) BYTE;
DECLARE X BASED YA BYTE,
        Y BASED YA BYTE,
        (XA,YA,CHECK1,CHECK2) ADDRESS,
        (XP,YP) BYTE;

/* EQUAL EXPONENTS */
XE=X(2) AND 80H;
YE=Y(2) AND 80H;
IF ( XE=YE ) THEN DO;
  XP=X(2) AND 7FH;
  YP=Y(2) AND 7FH;
  CHECK1=SHL(DOUBLE(X),8) OR X(1);
  CHECK2=SHL(DOUBLE(Y),8) OR Y(1);
  IF ( XE=80H ) THEN DO;
    IF ( XP<YP ) THEN RETURN 2;
    IF ( XP>YP ) THEN RETURN 0;
    IF CHECK2 < CHECK1 THEN RETURN 2;
    IF CHECK2 > CHECK1 THEN RETURN 0;
    RETURN 1;
  END;
  IF ( XP<YP ) THEN RETURN 0;
  IF ( XP>YP ) THEN RETURN 2;
  IF CHECK2 < CHECK1 THEN RETURN 2;
  IF CHECK2 > CHECK1 THEN RETURN 0;
  RETURN 1;
END;
IF ( XE = 0 ) THEN RETURN 2;
RETURN 0;

END COMPARE;
DECLARE
FUNCTION:
/* I=0
   I=1
   I=2
   I=3
   I=4 ATAN2
   DECLARE (E,V,D,V1,D1) (3) ADDRESS INITIAL(
   3000H,3300H,2D77H,
   3100H,3387H,2E3AH,
   3200H,346EH,2EFDH,
   0000H,3525H,0000H,
   0000H,35DCCH,0000H),
   I BYTE,
   LOOK (3) BYTE INITIAL (252,181,192),

```



```

MAX (3) BYTE INITIAL (252,181,192),
ENTA ADDRESS, ENT BASED ENTA BYTE,
DIFFA ADDRESS, VAL BASED VALA BYTE,
VALIA ADDRESS, DIFF BASED DIFFA BYTE,
VALIA ADDRESS, VALI BASED VALIA BYTE,
(LOOKI, MAXI) BYTE,
DIFFIA ADDRESS, DIFFI BASED DIFFIA BYTE;
/* INPUT CONSISTS OF A BASED VARIABLE,
IN GLOBAL VARIABLES I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z,
FOR RHO AND INVS RESPECTIVELY
DECLARE XA ADDRESS, X BASED XA BYTE;

ENTA=E(I); VALA=V(I); DIFFA=D(I);
VALIA=V(I); DIFFIA=D(I);
LCKKI=LOOK(I);
MAXI=MAX(I);
ENTRY=X; ENTRY=Y; ENTRY2=X(2);
IF(LCKKI=MAXI) THEN LOOKI=0;
ID=COMPARE(.ENTRY, ENTA+LOOKI);
IF(ID=1) THEN GO TO EXIT1;
/* READ UPWARDS IN THE TABLE*/
IF(ID>1) THEN DO;
IF(LOOKI=MAXI) THEN GO TO EXIT1;
ID=COMPARE(.ENTRY, ENTA+LOOKI);
IF(ID=1) THEN GO TO EXIT1;
IF(ID>1) THEN GO TO LOOP1;
LOOKI=LOOKI-3; GO TO EXIT2;
END;
/* READ DOWNWARDS IN THE TABLE */
LOOP2: IF(LOOKI=0) THEN GO TO EXIT1;
ID=COMPARE(.ENTRY, ENTA+LOOKI);
IF(ID=1) THEN GO TO EXIT1;
IF(ID>1) THEN GO TO LOOP2;
GOTO EXIT2;

EXIT1: T=VAL(LOOKI); T1=VAL(LOOKI+1); T2=VAL(LOOKI+2);
IF I<>1 THEN GO TO RET;
Z=VALI(LOOKI); Z1=VALI(LOOKI+1); Z2=VALI(LOOKI+2); GO TO RET;

EXIT2: TEMP=ENT(LOOKI); TEMP1=ENT(LOOKI+1); TEMP2=ENT(LOOKI+2);
/* CHANGE THE SIGN OF TEMP */
TEMP2=TEMP2 XOR 80H;
/* ENTRY - ENT(LOOK) */
CALL ADD(.TEMP, .ENTRY, .TEMP);
/* (ENTRY - ENT(LOOK))* DIFF(LOOK) */

```

OUTPUT WILL BE
Z, Z1, Z2,
*/


```

CALL MULT(.TEMP,DIFFA+LOOKI,.T);
CALL ADD(.T,VALA+LOOKI,.T);
/* VAL(LOOK)+ENTRY-ENT(LOOK) *DIFF(LOOK) */
IF I<>1 THEN GO TO RET;
CALL MULT(.TEMP,DIFFA+LOOKI,.Z);
CALL ADD(.Z,VALA+LOOKI,.Z);
RET: LOOK(I)=LOOKI;
RETURN;
END FUNCTION;
DECLARE (I,T1,T2,Z,Z1,Z2) BYTE; /* COS AND SIN */
CCSSIN: PROCEDURE(XA); COMPUTES BOTH THE SINE AND
/* THE COSINE FOR THE GIVEN ANGLE XA IN
THE COSINE THE OUTPUT VALUES ARE
STORED IN GLOBAL VARIABLES Z,Z1,Z2
FOR COSINE T,T1,T2 FOR SINE */
DECLARE (TH,TH1,TH2,THN,THN1,THN2,THN3,IMP1,IMP2) BYTE,
(MPI2,MPI21,MPI22) X BASED XA BYTE;
TH=X; TH1=X(1); TH2=X(2);
DECLARE OUT LABEL;
DECLARE I BYTE;
DO I=1 TO 4;
THN=TH; THN1=TH1; THN2=TH2;
CALL ADD (.TH,THN1,MPI2,IMP1,IMP2) /* TH IS THBAR - PI/2 */
IF (ZE AND 80H) <> 0 THEN GO TO OUT;
END;
RETURN;
OUT: TH2=ZE AND 7FH;
CALL FUNCTION (2,.THN);
THN=T; THN1=T1; THN2=T2;
/* THN= COS (TH) */
/* CALL FUNCTION (2,.TH); */
TH=T; TH1=T1; TH2=T2;
IF ((I=1) OR (I=3)) THEN DO;
T=THN; Z1=THN1; Z2=THN2;
IF (I=3) THEN DO;
Z2=Z2 XOR 80H;
T2=T2 XOR 80H;
RETURN;
END;
RETURN;
END
Z=TH; Z1=TH1; Z2=TH2 XOR 80H;

```



```

T=THN; T1=THN1; T2=THN2 ;
IF (I=4) THEN DO;
  T2=T2 XOR 80H;
  Z2=Z2 XOR 80H;
END;

RETURN;
END COSSIN;
SQRT: PROCEDURE (XA,PA);
/* ASSUME THAT XA IS A POSITIVE REAL NUMBER */
DECLARE XA ADDRESS, X BASED XA BYTE;
DECLARE PA ADDRESS, P BASED PA BYTE;
DECLARE (C,C1,C2,C3,B,B1,B2) BYTE;
/* INITIAL APPROXIMATION FOR THE ROOT IS
  EXP/2 */
DECLARE R BYTE;
B=X; B1=X(1);
B2=X(2)-64; IF
  (B2 AND 80H) =0 THEN B2=SHR(B2,1)+64;
ELSE DO; B2=>B2; B2=SHR(B2,1); B2=64-B2;
END;

DO R= 1 TO 4;
  CALL DIV(XA,B,C);
  CALL ADD(C,B,B);
  B(2)=B(2)-1;
END;
P(1)=B(1); P(2)=B(2);
RETURN;
END SQRT;
DECLARE COUNT BYTE INITIAL (0);
LN: A PROCEDURE (ARG,VAL);
/* IF X = 0 THEN CALCULATE THE VALUE OF LN(ABS(X)),
  REFERENCE THEN THE MAXIMUM NEGATIVE VALUE IS RETURNED. */
DECLARE ( ARG,VAL) ADDRESS, /* ARGUMENT AND ANSWER
  ADDRESSES */
X BASED ARG BYTE, Y BASED VAL BYTE, Z(3) BYTE, K BYTE;
DECLARE XE ADDRESS;
DECLARE (XA,ZA) ADDR(2**K-1) FOR K=0,...,8 */
/* VALUES OF LN(2**K) */
0B1H, 72H, 40H, /* LN(4/3) */
93H, 4BH, 3FH, /* LN(8/7) */
88H, 0BCH, 3EH, /* LN(16/15) */
84H, 02DH, 3DH, /* LN(32/31) */
82H, 0BH, 3CH, /* LN(64/63) */
81H, 03H, 3BH, /* LN(128/127) */
80H, 3FH, 3AH, /* LN(256/255) */
80H, 1FH, 39H, /* LN(512/511) */
, 80H, 10H, 37H, /* LN(1024/1023) */

```



```

OFFH,0F0H,32H, /* LN(2048/2047) */
OFFH,0F8H,34H /* LN(4096/4095) */
);
/* CHECK FOR A ZERO ARGUMENT */
IF X=0 THEN DO;
Y=OFFH; Y(1)=OFFH; Y(2)=OFFH; RETURN; END;
/* ARG := ABS ARG */
XE=(DOUBLE(X(2))) AND 7FH -41H;
CALL FLGAT(XE,Z);
CALL MUL(Z,CONS,VAL);
XA=X; XA=SHL(XA,8); XA= XA OR X(1);
K=1;
ZA=SHR(XA,1);
DO WHILE((XA <> 8000H) AND (XA > ZA));
IF (XA < (ZA OR 8000H)) THEN /* XA-ZA < 1 */
DO;
ZA= SHR(ZA,1);
IF (K=13) THEN DO; COUNT = COUNT+1;
RETURN; END;
K=K+1;
END;
ELSE
DO;
XA=XA-ZA;
Y = Y + LN(2**K/(2**K-1)) /*
CALL ADD(VAL,CONS(3*(K-1)),Z);
Y=Z; Y(1)=Z(1); Y(2)=Z(2);
ZA= SHR(XA,K);
END;
END /* END OF DO WHILE */;
RETURN;
END LN;
SAMP: PROCEDURE;
DECLARE (IAS,RATE) ADDRESS, CSSW BYTE;
DECLARE AGF (3) BYTE;
DECLARE THOU(3) BYTE INITIAL(OFAH,00H,4AH);
DECLARE
(AZR,PR,TOIR,SINP,COSP,K,VM) (3) BYTE,
(ALT) ADDRESS,
(THET(3) BYTE, (L,SHIFTD,PL)(3) BYTE,
(PLEAD,ALRAD) BYTE;
DECLARE RANGE(3) BYTE;
DECLARE (VF,VC)(3) BYTE;
DECLARE RHO(78) BYTE INITIAL(
/* ALTITUDE RHO */
/* 0 */ 99H,0E1H,3DH,
/*2048*/ 90H,3EH,3DH,
/*4096*/ 87H,35H,3DH,
/*6144*/ 0FDH,7BH,3CH,

```



```

/*8192*/ OEDH,9BH,3CH,
/*10240*/ ODEH,0BAH,3CH,
/*12288*/ ODOH,0C7H,3CH,
/*14336*/ OC3H,0B3H,3CH,
/*16384*/ OB7H,72H,3CH,
/*18432*/ OABH,0F5H,3CH,
/*20480*/ OA1H,30H,3CH,
/*22528*/ O97H,17H,3CH,
/*24576*/ O8DH,0A1H,3CH,
/*26624*/ O84H,0C2H,3CH,
/*28672*/ OF8H,0E3H,3BH,
/*30720*/ OE9H,4DH,3BH,
/*32768*/ ODAH,0B1H,3BH,
/*34816*/ OCCD,0FEH,3BH,
/*36864*/ CCOH,48H,3BH,
/*38912*/ OB4H,1FH,3BH,
/*40960*/ OA8H,0D7H,3BH,
/*43008*/ 9EH,44H,3BH,
/*45056*/ 94H,5AH,3BH,
/*47104*/ 8BH,20H,3BH,
/*51200*/ OF4H,61H,3AH)
KDP (3) BYTE INITIAL(0B0H,98H,38H),
      GDIV2 (3) BYTE INITIAL(0D8H,31H,41H),
      (TEMP1,TEMP2,TEMP3,TEMP4,TEMP5,TEMP6,TEMP7,TEMP8,TEMP16,
      TEMP9,TEMP10,TEMP11,TEMP12,TEMP13,TEMP14,TEMP15,TEMP16,
      TEMP17,TEMP18,TEMP19,TEMP20) (3) BYTE;
DECLARE DUM (3) BYTE;
DECLARE KONS (3)
GUNS: PROCEDURE;
CALL ADD(.TEMP1,DUM,.VM); /* VM=3300+VF*1.689 */
CALL MUL(.VM,KONS,.TEMP1); /* TEMP1=VM*KONS */
CALL SUB(.VF,.VC,.TEMP3); /* TEMP2=TEMP1+1 */
CALL LN(.TEMP2,.TEMP4); /* TEMP4=LN(VM*KONS+1) */
CALL DIV(.TEMP4,.KONS,.TEMP5); /* LN(VM*KONS+1)/KONS */
CALL ADD(.TEMP3,.TEMP5,.TEMP8); /* RANGE+(VF-VC) */
CALL SUB(.TEMP8,.TEMP5,.TEMP9); /* F(1.0) */
CALL DIV(.VM,.TEMP10); /* VM/(VM*KONS+1) */
CALL SUB(.TEMP9,.TEMP11); /* F(1.0) */
CALL DIV(.TEMP9,.TEMP12); /* F(T)/F(T) */
CALL SUB(.ONE,.TEMP13,.TEMP14); /* TIME OF FLIGHT */
CALL MUL(.TEMP1,.TEMP13,.TEMP14); /* VM*KONS*TIME+1 */
CALL ADD(.ONE,.TEMP14,.TEMP15); /* LN(VM*KONS*TIME+1) */
CALL LN(.TEMP15,.TEMP16); /* LN(VM*KONS*TIME) */
CALL MUL(.KONS,.TEMP13,.TEMP17); /* KONS*TIME */
CALL DIV(.TEMP16,.TEMP17,.TEMP18); /* AVERAGE VELOCITY

```



```

TAKEN AS LN(VM*KONS*TIME+1)/KONS*TIME */
CALL MUL(.TEMP13,.TEMP13,.TEMP16); /* TIME**2 */
CALL MUL(.TEMP16,.THOU,.TEMP17); /* TIME**2 *1000 */
CALL MUL(.TEMP13,.TEMP18,.TEMP4); /* RANGE TO IMPACT */
CALL MUL(.GOIV2,.TEMP17,.TEMP19); /* GRAVITY DROP */
CALL DIV(.TEMP19,.TEMP4,.TEMP20); /* MILS GRAVITY DROP */
CALL DIV(.VF,.VM,.TEMP1);
CALL MUL(.TEMP1,.AGF,.SHIFT); /* LEAD FOR TRAJECTORY
SHIFT */
CALL MUL(.TOIR,.RANGE,.TEMP2); /* TGT MOTION NORMAL */
CALL DIV(.TEMP2,.TEMP18,.TEMP3); /* LEAD FOR TGT MOTION
*/
/* LEAD IN PITCH = LEAD FOR TGT MOTION *
(SIN* COS(THET))+COSP*SIN(THET))+TRAJECTORY SHIFT
+3.9*GRAVITY DROP*COS(THET) */
CALL MUL(.Z,.COSP,.TEMP4); /* SIN(THET)*COSP */
CALL MUL(.Z,.SINP,.TEMP5); /* COS(THET)*SINP */
CALL ADD(.TEMP4,.TEMP5,.TEMP6);
CALL MUL(.TEMP3,.TEMP6,.TEMP7); /* L*SIN(THET+PHI) */
CALL ADD(.TEMP7,.SHIFT,.TEMP4); /* GRAVITY DROP */
CALL MUL(.Z,.TEMP20,.TEMP7);
CALL ADD(.TEMP4,.TEMP7,.TEMP5);
CALL FIX(.TEMP5,.PLEAD);
PLEAD=PLEAD+4; /* PITCH LEAD */
CALL MUL(.Z,.COSP,.TEMP1); /* COS(THET)*COS(PHI) */
CALL MUL(.SINP,.TEMP2); /* - SIN(THET)*SIN(PHI) */
TEMP2(2)=TEMP2(2)+XOR 80H; /* L* COS(THET+PHI) */
CALL ADD(.TEMP1,.TEMP2,.TEMP4);
CALL MUL(.TEMP4,.TEMP3,.TEMP6); /* L* COS(THET+PHI) */
CALL ADD(.TEMP6,.TEMP20,.TEMP8);
CALL MUL(.TEMP20,.Z,.TEMP5);
CALL ADD(.TEMP8,.TEMP5,.TEMP6); /* AZMUTH LEAD */
CALL FIX(.TEMP6,.ALEAD);
AA2:
RETURN;
END GUNS;
/* SIMULATED SAMPLING LOOP */
/* READ ALT */
ALT=100;
/* READ TAS */
TAS=500;
CALL FLOAT(.TAS,.VF);
/* READ BANK ANGLE */
THET=0; THET(1)=0; THET(2)=0;
CALL COS(SIN(THET));
/* READ CLOSING VELOCITY */
TAS=50;
CALL FLOAT(.TAS,.VC);

```



```

/*- READ RANGE TO TGT */
DECLARE FEET ADDRESS;
FEET=1000;
CALL FLOAT(.FEET,.RANGE);
/* READ PITCH RATE */
RATE=120;
CALL FLOAT(.RATE,.AZR);
CALL DIV(.AZR,.THOU,.PR);
/* READ AZIMUTH RATE */
RATE=10;
CALL FLOAT(.RATE,.AZR);
DECLARE TEMR (3) BYTE;
CALL FLOAT(.RATE,.TEMR);
CALL DIV(.TEMR,.THOU,.AZR);
/* READ ANGLE OF ATTACK IN MILS */
TD=150;
CALL FLOAT(.TD,.AGF);
/* SAMPLE SWITCH */
CSSW=OFFH;
CALL MUL(.AZR,.AZR,.TEMP1);
CALL MUL(.PR,.PR,.TEMP2);
CALL ADD(.TEMP1,.TEMP2,.TEMP3);
CALL Sqrt(.TEMP3,.TOTR);
/* TOTAL ANGULAR RATE = Sqrt(PITCH RATE **2 +
AZIMUTH RATE **2) */
CALL DIV(.PR,.TOTR,.SINP);
CALL DIV(.AZR,.TOTR,.COSPP);
CALL MUL(.RHO(SHR(ALT,11)-1), .KDP,.KONS);
CALL MUL(.VF,.FPS,.TEMP1);
AA4:
IF (CSSW) THEN CALL GUNS;
ELSE RETURN;
END SAMP;

CALL SAMP;
DECLARE JII ADDRESS INITIAL(10);
DECLARE (XJJ,YJJ,ZJJ) (3) BYTE,(JX,JY,JZ) ADDRESS;
JX=85; JY=91; JZ=98;
CALL FLOAT(.JX,.XJJ); CALL FLOAT(.JY,.YJJ);
AA3:
CALL FLOAT(.JZ,.ZJJ);
AA6:
DECLARE (DEN,TEM)(3)BYTE;
BB6:
CALL FLOAT(.JII,.DEN);
AA5:
CALL DIV(.XJJ,.DEN,.TEM);
BB1:
CALL LN(.TEM,.XJJ);

```



```

BB2: CALL DIV(.YJJ,.DEN,.TEM);
BB3: CALL LN(.TEM,.YJJ);
BB4: CALL DIV(.ZJJ,.DEN,.TEM);
BB5: CALL LN(.TEM,.ZJJ);
AA1:
  JII=1;
  EOF
//PASS2.CARD DD *
$F=1 $G=1 $M=1 $Q=29 $B=7
//INTERP.CARD DD *
$F=1
LOAD 7 7.
$P=1
$MAX = 100000
DISPLAY SYMBOLS.
REFER AA1,AA2,AA3,AA4,AA5,AA6.
REFER BB1,BB2,BB3,BB4,BB5,BB6.
BASE HEX.
GO.
  TIME.
GO.
  TIME.
DIS MEM ALEAD.
DIS MEM PLEAD.
$EOF

```


LIST OF REFERENCES

1. Gauthier, D. P., Effects of Short Period Angle of Attack Oscillation on Air-to-Ground Weapons Dive Delivery Accuracy, Master's Thesis, Naval Postgraduate School, June 1974.
- i. NAVAIR 01-245FDB-1T, Tactical Manual Navy Model F-4B, F-4J, F-4N Aircraft, Commander Operational Test and Evaluation Force, September 1972.
3. USAF Fighter Weapons School Instructional Text Non-Nuclear Weapons Delivery, Chief of Academics 57th Fighter Weapons Wing (TAC), January 1974.
4. NAVAIR 01-245FDB-1, Natops Flight Manual, Navy Model, F-4B and F-4N Aircraft, Commander Naval Air Systems Command, February 1973.
5. Conte, S. D. and de Boor, C., Elementary Numerical Analysis, 2d ed., Mc Graw-Hill, 1972.
6. NAVAIR 01-245FDB-2-4.2, Technical Manual Maintenance Instructions Organizational and Intermediate, Air Data Computer Set, Navy Model, F-4B Aircraft, Commander, Naval Air Systems Command, April 1972.
7. Naval Weapons Laboratory Technical Memorandum No. K72/65. Stretch Computer Program to

Integrate Particle Trajectories Using Navy
Standard Atmosphere, by Kay B. Pollock, pp 4,5
May 1965.

8. USAF Fighter Weapons School Instructional Text,
Rear Hemisphere Aerial Attack Systems, Chief of
Academics 57th Fighter Weapons Wing (TAC),
February 1971.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Documentation Center 2
Cameron Station
Alexandria, Virginia 22314
2. Library, Code 0212 2
Naval Postgraduate School
Monterey, California 93940
3. Chairman, Code 72 1
Computer Science Group
Naval Postgraduate School
Monterey, California 93940
4. Professor U. R. Kodres, Code 72Kr 1
Computer Science Group
Naval Postgraduate School
Monterey, California 93940
5. MAJOR James Pennell, USMC 1
Call Hollow Road
Pomona, New York 10970

Thesis
P3297 Pennell
c.1

161216

An investigation of
the use of micro
computers to aid the
pilot in the solution
of the ballistics
problem.

21 OCT 77
24 JAN 79
23 JAN 80

23120
24656
25232
26005
27070

Thesis
P3297 Pennell
c.1

161215

An investigation of
the use of micro
computers to aid the
pilot in the solution
of the ballistics
problem.

thesP3297

An investigation of the use of micro com



3 2768 001 97956 0
DUDLEY KNOX LIBRARY